microDXP

Digital X-Ray Processor

Technical Reference Manual

Version 1.1.2

August 6, 2009

MicroDXP Hardware Revision: E

Micromanager Software Revision: 2.3.x

XIA LLC

31057 Genstar Rd.
Hayward, CA 94544 USA
Tel: (510) 401-5760; Fax: (510) 401-5761
http://www.xia.com/

Safety

Please take a moment to review these safety precautions. They are provided both for your protection and to prevent damage to the Micro Digital X-ray Processor (μDXP) and connected equipment. This safety information applies to all operators and service personnel.

Power Source

The microDXP requires several DC voltage supplies to operate. In such cases where the user will provide their own power supplies, they must conform to the specifications contained in section Appendix D of this manual to avoid damaging the microDXP and connected equipment, and nullifying the product warranty.

We recommend that new users purchase either the microDXP RS232 Rapid Development Kit or the microDXP USB Rapid Development Kit. The RS232 kit includes a wall-mount power supply intended to operate from a mains supply voltage of 120VAC at 60Hz. The included microCOM interface circuit board provides linear regulation of the DC voltages required by the microDXP.

The USB kit, introduced August 2009, includes a wall-mount power supply that can accommodate 100VAC to 240VAC at 47Hz to 63Hz. In addition, it can adapt to plug styles for North America, Japan, Europe, UK, and Australia. The included MicroComU board generates the various DC voltages required by the microDXP.

Use of either evaluation kit with any other mains voltage or power supply could damage the unit and nullify the product warranty. Refer to the **Rapid Development Kit Manual** for instructions on installing the power supply.

Detector and Preamplifier Damage

The microDXP input impedance is 1kΩ, and should be compatible with most preamplifiers. Please consult the documentation provided by the preamplifier manufacturer to confirm that such a load is acceptable.

Because the microDXP does not provide power for the detector or preamplifier there is little risk of damage to either resulting from the microDXP itself. Nonetheless, please review all instructions and safety precautions provided with these components before powering a connected system.

Servicing and Cleaning

To avoid personal injury, and/or damage to the microDXP or connected equipment, do not attempt to repair or clean the unit. The microDXP hardware is warranted against all defects for 1 year. Please contact the factory or your distributor before returning items for service.

End Users Agreement

XIA LLC warrants that this product will be free from defects in materials and workmanship for a period of one (1) year from the date of shipment. If any such product proves defective during this warranty period, XIA LLC, at its option, will either repair the defective products without charge for parts and labor, or will provide a replacement in exchange for the defective product.

In order to obtain service under this warranty, Customer must notify XIA LLC of the defect before the expiration of the warranty period and make suitable arrangements for the performance of the service.

This warranty shall not apply to any defect, failure or damage caused by improper uses or inadequate care. XIA LLC shall not be obligated to furnish service under this warranty a) to repair damage resulting from attempts by personnel other than XIA LLC representatives to repair or service the product; or b) to repair damage resulting from improper use or connection to incompatible equipment.

THIS WARRANTY IS GIVEN BY XIA LLC WITH RESPECT TO THIS PRODUCT IN LIEU OF ANY OTHER WARRANTIES, EXPRESSED OR IMPLIED. XIA LLC AND ITS VENDORS DISCLAIM ANY IMPLIED WARRANTIES OF MERCHANTABILITYOR FITNESS FOR A PARTICULAR PURPOSE. XIA LLC'S RESPONSIBILITY TO REPAIR OR REPLACE DEFECTIVE PRODUCTS IS THE SOLE AND EXCLUSIVE REMEDY PROVIDED TO THE CUSTOMER FOR BREACH OF THIS WARRANTY. XIA LLC AND ITS VENDORS WILL NOT BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IRRESPECTIVE OF WHETHER XIA LLC OR THE VENDOR HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

Contact Information:

XIA LLC
31057 Genstar Rd.
Hayward, CA 94544 USA
microDXP@xia.com (for microDXP, microCOM, or MicroComU hardware support)
software_support@xia.com (for microManager software support)

(510) 401-5760

Manual Conventions

Through out this manual we will use the following conventions:

Convention	Description	Example
»	The » symbol leads you through nested menu items, DAQExplorer items, and dialog box options.	The sequence **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and choose **Options** from the sub menu.
Bold	Bold text denotes items that you must select or click on in the software, such as menu items, and dialog box options.	...expand the **Run Control** section of the **DAQExplorer** to access the run presets.
[Bold]	Bold text within [ ] denotes a command button.	**[Start Run]** indicates the command button labeled Start Run.
`monospace`	Items in this font denote sections of code, file contents, and syntax examples.	`Setup.exe` refers to a file called "setup.exe" on the host computer.
"window"	Text in quotation refers to window titles, and quotations from other sources	"Options" indicates the window accessed via **Tools»Options**.
Italics	Italic text denotes a new term being introduced , or simply emphasis	*peaking time* refers to the length of the slow filter. ...it is important first to set the energy filter Gap so that SLOWGAP to *at least one unit greater than* the preamplifier risetime...
<Key> <Shift-Alt-Delete> or <Ctrl+D>	Angle brackets denote a key on the keybord (not case sensitive). A hyphen or plus between two or more key names denotes that the keys should be pressed simultaneously (not case sensitive).	<W> indicates the W key <Ctrl+W> represents holding the control key while pressing the W key on the keyboard
Bold italic	Warnings and cautionary text.	***CAUTION:*** *Improper connections or settings can result in damage to system components.*
CAPITALS	CAPITALS denote DSP parameter names	SLOWLEN is the length of the slow energy filter

1 Introduction

The Micro Digital X-ray Processor (microDXP) is a high rate, digitally-based, multi-channel analysis spectrometer designed for energy dispersive x-ray or γ-ray measurements in benchtop, networked, portable and embedded systems. Its versatile analog front-end accommodates most solid-state and gas detectors and a wide range of common preamplifiers, including pulsed optical reset, transistor reset, and resistive feedback types. The microDXP offers complete computer/PDA control over all available amplifier and spectrometer controls including gain (if applicable), filter peaking time, and pileup inspection criteria. As with all DXP instruments, the firmware (i.e. DSP, Microcontroller and FPGA code) can be upgraded in the field. Unlike other instruments in the DXP family, microDXP firmware and parameters are stored locally in non-volatile memory. The architecture supports custom auxiliary digital access including the industry-standard I^2C serial bus and four configurable digital I/O lines. Finally, *the microDXP consumes less than 600mW of power*. The microDXP is thus a flexible, cost-effective OEM component that can form the core of a broad range of systems, from basic XRF to the most demanding process and control applications emerging in research and industry.

1.1 MicroDXP Overview

The microDXP offers a set of standard and customizable features and options intended to address the design requirements of a wide range of complex spectroscopy data acquisition systems. Application examples are given below in §1.1.3.

1.1.1 Features

The microDXP is the latest addition to the Digital X-ray Processor (DXP) line of instruments from XIA. The DXP is a digitally based spectrometer architecture that offers general control over all analog and digital settings as outlined in §1.1.1.1 below. A unique feature of the microDXP is non-volatile parameter storage on a per-peaking-time basis. For each **Peaking Time** the DSP stores a complete set of all related spectrometer parameters into non-volatile memory, as outlined in §1.1.1.2. Five **Peaking Times** are available for each FiPPI decimation (each decimation provides a range of peaking times) included in the purchase, yielding a maximum of fifteen (15) independently optimized **Peaking Times**. Re-selecting an optimized **Peaking Time** conveniently retrieves all relevant digital filtering and peak inspection parameters (e.g. gap time, thresholds, pileup inspection interval, etc.) from non-volatile flash memory. This functionality is particularly beneficial in embedded systems: The configuration and optimization procedures can be delegated to the development kit platform (i.e. the MicroComU companion board and MicroManager software), leaving a very tight data acquisition command set for the embedded system itself. The configuration process need only be performed once, though of course the parameter sets be modified at any time.

Under NRE terms, XIA develops customized firmware to support specialized data acquisition modes, and to support auxiliary hardware functions (see §1.1.1.3 Custom OEM Features). The I^2C bus can be used to control various devices, e.g. x-ray tubes and DACs, and the four auxiliary lines can be individually configured as inputs (e.g. to accept a synchronization signal in a time-resolved spectroscopy experiment) or as outputs (e.g. multiple SCAs).

1.1.1.1 General Spectrometer Features

- Extremely compact unit replaces spectroscopy amplifier, shaping amplifier and multi-channel analyzer (MCA) at significantly reduced cost and power consumption.
- Operates with a wide variety of x-ray or γ-ray detectors using preamplifiers of pulsed optical reset, transistor reset or resistor feedback types.
- Multi-channel analysis with up to 8K (8132) bins, allowing for optimal use of data to separate fluorescence signal from backgrounds.
- Instantaneous throughput up to 164,000 counts per second (cps) in the spectrum for the standard 8MHz pipeline clock speed; up to 328,000 cps for the 16MHz pipeline clock option (see §1.1.2.3).
- Digital trapezoidal filtering, with programmable peaking times between 750 ns and 48 μs for the standard 8MHz pipeline clock speed, and between 375 ns and 24 μs for the 16MHz pipeline clock option (see §1.1.2.3).
- Digitally-controlled internal gain (requires variable gain option, see §1.1.2.5): 32dB range, 16 bit precision.
- Pileup inspection criteria computer settable, including fast channel peaking time, threshold, and rejection criterion.
- Accurate ICR and live-time reporting for precise dead-time corrections.

1.1.1.2 Embedded Systems Features

- Up to fifteen (15) sets of optimized spectrometer parameters can be stored, and later retrieved with a single command.
- Five (5) sets of MCA parameters, or spectrum formats, can be stored, and later retrieved with a single command. *Note:* The microManager software currently supports only one MCA set.
- XIA provides the microManager software application for microDXP parameter set configuration—intended for use both in evaluation and production phases.
- A small and powerful RS-232 data acquisition command set.

1.1.1.3 Custom OEM Features

- Customized firmware development for special applications such as time-resolved spectroscopy, multiple SCA's and pulse-shape analysis.

- Flexible auxiliary digital I/O: 4 general purpose lines, a Gate signal to externally control data acquisition, the (emerging) industry-standard I^2C bus interface and an external interrupt line.
- Several assembly options are offered that exclude various hardware and software features in order to reduce the cost for dedicated applications (further details in the following section).

1.1.2 Options and Specifications

***Note:** The hardware options included with your microDXP are specified on the information sheet included with the shipment.*

The microDXP hardware is both powerful and flexible. In many cases, particularly for dedicated embedded systems, much of the microDXP's resources will not be used. XIA is pleased to offer a number of assembly options that will produce the best value for a given set of requirements. The specification sheet provided with your microDXP lists the options that have been implemented. *Note:* The firmware options and clock speed can be upgraded in the field, whereas the power supply and gain options require physical modifications to the hardware. Please contact XIA at microDXP@xia.com for more information about hardware options and upgrades.

1.1.2.1 Communications and Power Interface

The microDXP hardware offers three communications options via two high-density connectors: a flat-flex cable for low and medium speed serial communications and a board-to-board connector that offers both serial protocols plus high-speed parallel access. In all cases the power, communications and auxiliary digital I/O is carried on a single connector, and the analog input signal enters via a separate connection to ensure immunity from electro-magnetic interference.

The standard assembly offers RS -232 communications and auxiliary I/O via the flex-cable interconnect. RS-232 runs at 115kbaud, with burst rates up to 10 Kbytes/sec.

The flex interconnect also supports Analog Devices DSP serial port (SPORT) communications for faster data transfers, up to 2 Mbytes/sec. This option is targeted for multi-channel systems and will require some user hardware and DSP code development.

A third communications option offers parallel IDMA access to DSP memory for transfer rates up to 10 Mbytes/sec. This interface is used as the basis for the new MicroComU companion board and microDXP USB Rapid Development Kit.

1.1.2.2 Power Supplies

Two power supply variants are available, corresponding to whether on-board regulators for the analog supply voltages are used or are bypassed. The analog circuitry requires +/-5.0V, either supplied directly or indirectly through on-board LDO regulators. Excessive voltage spikes and/or ripple on the analog supplies, >20mVpp, can seriously degrade system performance. If +/-5.0V is supplied directly, either linear regulated or high-quality switching supplies should be used. If the on-board LDO regulators are used a minimum of +/-5.50V is required, and the ripple requirement can be relaxed a bit.

If planning to use the MicroComU companion board, choose the variant of the microDXP that uses its on-board voltage regulators. Aside from this, the MicroComU board will take care of generating all required voltages for the microDXP, at the specified currents and noise performance.

Regulated Supply Option: (<20mV pk-pk noise)			
Voltage Range	*Current (min)*	*Current (max)*	*Description*
+3.3V +/- 150mV	100mA	130mA	Decent switching supply
+5.0V +/- 100mV	25mA	30mA	Linear or high-quality switching
-5.0V +/- 100mV	25mA	30mA	Linear or high-quality switching
Unregulated Supply Option (<100mV pk-pk noise)			
Voltage Range	*Current (min)*	*Current (max)*	*Description*
+3.3V +/- 150mV	100mA	130mA	Decent switching supply
+5.5V to +6.0V	25mA	35mA	Decent switching supply
-5.5V to –6.0V	25mA	35mA	Decent switching supply

Table 1.1: Power supply options and specifications for the microDXP.

For both variants, the onboard digital circuitry draws from a 3.3V supply input, with on-board regulators employed to produce 2.5V for the DSP and 3.0V for the ADC. The ripple requirements for this supply are not particularly stringent, though excessive radiated noise is to be avoided. If a switching supply is used, it should be well shielded from, and properly grounded with respect to, the microDXP.

1.1.2.3 Pipeline Clock Speed Choice

The DSP operates with a 32MHz clock, and a clock-divider produces either 16MHz or 8MHz for the more power-hungry ADC and the FiPPI (Filter-Pulse-Pileup-Inspector). Simply put, the clock speed determines the scale of available peaking times (see Table 1.3 below), with a faster clock producing shorter peaking times and thus a higher output count rate, or OCR. Note however that a higher clock speed will result in slightly increased power consumption. The following table illustrates these points:

Clock Speed	Sampling Period	Maximum OCR	Power Consumption
8 MHz	125 ns	164,000 cps	514 mW
16 MHz	62.5 ns	328,000 cps	558 mW

Table 1.2: Data pipeline (ADC and FiPPI) clock speed options.

1.1.2.4 Firmware Selection

The term firmware refers both to code running on the DSP and the FPGA configuration code that comprises the Filter-Pulse-Pileup-Inspector, or FiPPI. Updates to both the DSP and FPGA codes will be posted to the XIA website. Please check the microDXP page at:

http://www.xia.com/microDXP.html

The on-board digital signal processor (DSP) monitors the analog circuitry, manages spectrum scaling and binning, and carries out various high-level calculations. The standard DSP code included with all orders will be sufficient to satisfy most users' requirements. In some cases custom DSP code may be provided for specialized applications.

The reconfigurable digital shaping, triggering, and pileup-rejection algorithms are contained in a field-programmable-gate-array (FPGA). Each complete configuration code is referred to as a FiPPI (**Fi**lter **P**ulse **P**ileup **I**nspector). FiPPIs are denoted by their decimation. Each FiPPI uses a decimator circuit to pre-average the ADC codes before trapezoidal shaping is applied. The so-called decimation determines the peaking time range supported by a particular FiPPI. The microDXP hardware can store between one and three FiPPI decimations, allowing great flexibility in choosing the proper peaking time range at the lowest cost. Table 1.3 lists the peaking times offered for each FiPPI decimation; the peaking time range of course depends on the chosen pipeline clock speed.

FiPPI Decimation	#ADC codes pre-averaged	Peaking times available with **8MHz** clock	Peaking times available with **16MHz** clock
0	0	0.75, 1.125, 1.5, 2.25, 3	0.375, 0.5625, 0.75, 1.125, 1.5
1	2	1.5, 2.25, 3, 4.5, 6	0.75, 1.125, 1.5, 2.25, 3
2	4	3, 4.5, 6, 9, 12	1.5, 2.25, 3, 4.5, 6
3	8	6, 9, 12, 18, 24	3, 4.5, 6, 9, 12
4	16	12, 18, 24, 36, 48	6, 9, 12, 18, 24

Table 1.3: Firmware peaking time range options.

For each FiPPI decimation five sets of DSP parameters, or PARSETs, are stored in non-volatile memory corresponding to the five available peaking times. This allows for the procedural separation of parameter optimization and data acquisition: During the setup process the parameters for each peaking time are optimized once and saved, and are then automatically retrieved whenever the peaking time is selected for acquisition. Parameter set storage and retrieval is described in further detail the Getting Started section below.

1.1.2.5 Gain and Calibration Options

Two gain configurations are available as hardware options. With the fixed-gain option, a user-selected analog gain is implemented in the on-board circuitry. The gain tolerance will typically be at the one percent level; typically an energy calibration must be handled offline by the host software.

The variable gain option provides 32dB of 16-bit precision, digitally controlled gain. This allows the microDXP to be optimized for a wide range of x-ray energies, and allows for a real hardware calibration.

1.1.3 Application Examples

The microDXP miniaturized circuit-board can easily be incorporated into a variety of benchtop, portable, networked and embedded x-ray and γ-ray spectroscopy data acquisition systems. In the first example below, the microDXP / MicroComU board set runs on a laboratory benchtop under the control of an x86 Personal Computer. No user hardware design is required, no power supplies are needed, and no microDXP hardware or firmware modifications are necessary.

In the second example below, the microDXP runs either on the laboratory benchtop as a peripheral device under the control of an x86 Personal Computer, or similarly in portable systems under the control of a PDA. Minimal user hardware design, and no microDXP hardware or firmware modifications are required.

In the third example a more complex dedicated system is considered. The I^2C serial bus is used to control a ‘smart’ x-ray tube and detector HV bias supply, and the auxiliary digital I/O drives electromechanical or pneumatic components in real time based upon user defined metrics of acquired data.

1.1.3.1 Example 1. General-Purpose Spectrometer Using USB 2.0 and MicroComU Companion Board

In this example the microDXP / microComU board set acts as a general-purpose spectrometer, connected as a peripheral device under the control of a host computer. No specialized data acquisition modes are required, thus no firmware development is necessary. No user hardware design is required. All required power is drawn from the USB port.

XIA non-recurring engineering (NRE) required: NONE.

User development required: ALMOST NONE.

1. It may be necessary to design an enclosure.

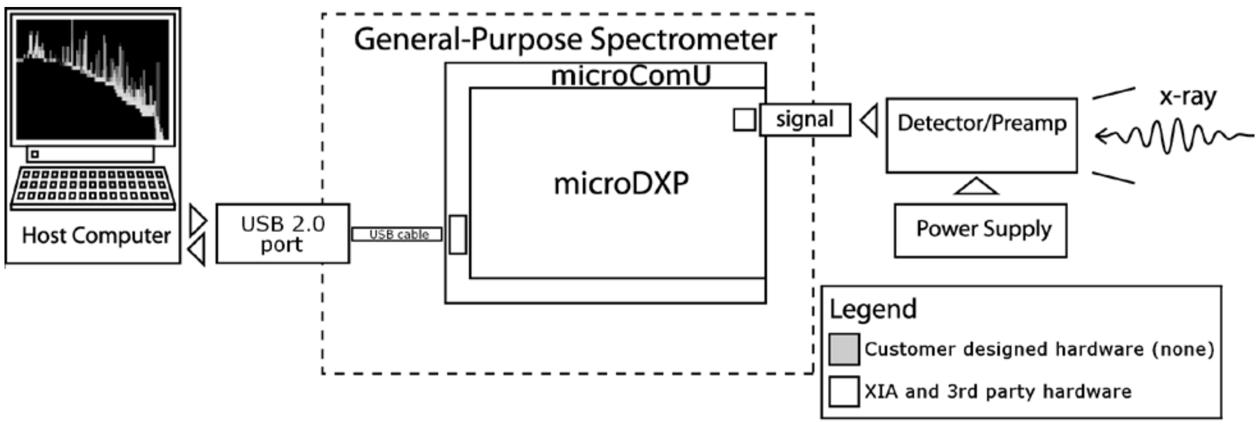

Figure 1.1: A general-purpose spectrometer incorporating the microDXP and MicroComU companion board. The board set communicates with a host PC using USB 2.0. The board set derives all necessary power supplies from the USB port.

In this example, the MicroComU board acts as both a carrier and companion board for the microDXP. Power for the MicroComU/microDXP board set may be taken from the USB port as shown in the example above, or may be provided externally, as with the AC wall adapter provided with the USB Rapid Development Kit. MicroComU board dimensions and mounting information, the connector locations and specifications, and the power supply specifications are all found in the separate **MicroComU Technical Reference Manual** available at:

http://www.xia.com/microDXP.html

1.1.3.2 Example 2. General-Purpose Spectrometer Using RS232 and Custom Breakout Board

In this example the microDXP acts as a general-purpose spectrometer, connected as a peripheral device under the control of a host computer or PDA. No specialized data acquisition modes are required, thus no firmware development is necessary. Some user hardware design is, however, required.

XIA non-recurring engineering (NRE) required: NONE.

User development required:

1. To connect to the external host, only a simple *routing adapter* interface unit is required to break out the microDXP high-density internal connection to standard RS-232 and power connections. At a minimum, this interface is a wire harness but could entail a printed circuit board with a small number of passive components.
2. Power supplies for the microDXP must be provided. Optional voltage regulators for the analog circuitry are included on the microDXP for systems in which high-quality power supplies are not available.
3. Some additional mechanical design, i.e. enclosure design, may be necessary.

MicroDXP board dimensions and mounting information, the connector locations and specifications, and the power supply specifications are all found in Appendix D of this manual.

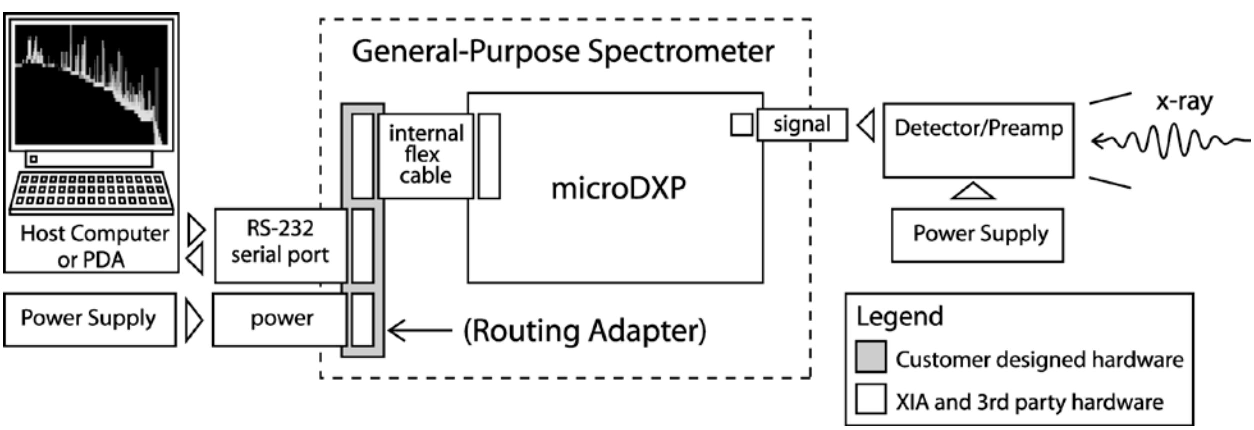

Figure 1.2: A general-purpose spectrometer incorporating the microDXP. A simple user-designed routing-adapter interface connects the microDXP to the host computer/PDA and power supplies.

The microDXP, power supplies and 'routing-adapter' together constitute a spectrometer that can be connected to virtually any controller with RS-232 communications. *Note:* The microCOM interface board included with the RS232 Rapid Development Kit falls into this category, though its large form-factor nearly prohibits commercial use. However, the recently released

MicroComU board (included with the USB Rapid Development Kit) may also be used for this purpose.

1.1.3.3 Example 3. Dedicated Spectrometer Using RS232

This example considers a materials sorting application where objects with certain pre-defined alloy ratios X, Y and Z are to be separated from others. An x-ray source irradiates incoming samples, and incident x-rays are collected by a solid-state detector connected to the microDXP. The microDXP is configured to assert a combination of its auxiliary digital I/O lines whenever the peak ratio X,Y, or Z is detected. The digital I/O lines drive electromechanical or pneumatic components in real-time to execute the appropriate mechanical operation, e.g. put the recognized object in the desired bin. User controls are limited to starting and stopping the system, and selecting one out of a small number of operating modes. Power supplies for the microDXP are also included. Finally, an external data port (e.g. RS-232) is also included so that ratios corresponding to new alloys can be defined, and new firmware uploaded without dismantling the hardware; or, alternatively, the microDXP could periodically be run in full MCA mode under computer control for diagnostic purposes.

This example demonstrates a system that uses a very small data acquisition command set (i.e. ‘start run’ and ‘stop run’) but that, conversely, requires customizations to the microDXP as well as significantly more user-designed hardware.

XIA non-recurring engineering (NRE) required:

1. Customized PIC microcontroller code is required to implement the I^2C peripheral device control.
2. Customized PIC microcontroller code is required to implement high-level data acquisition routines controlled through the user pushbutton interface.
3. Customized DSP code is required for peak ratio calculations, possibly implemented in lookup tables.
4. Minimal FiPPI (FPGA) code modification is required to implement the auxiliary digital I/O functionality.

User development required:

1. A more advanced interface unit is required to break out the microDXP high-density internal connection to standard RS-232, auxiliary and power connections. Still, this interface does not involve many active components, i.e. the I^2C and auxiliary digital I/O are simply routed to additional connectors. The pushbutton interface might include an additional microcontroller, but could be implemented simply in logic.
2. As drawn, the power supply is integrated on the interface board, with the same requirements as in the previous example. As stated there, optional voltage regulators for the analog circuitry are included on the microDXP for systems in which high-quality power supplies are not available.

3. Again, some additional mechanical design, i.e. enclosure design, may be necessary.

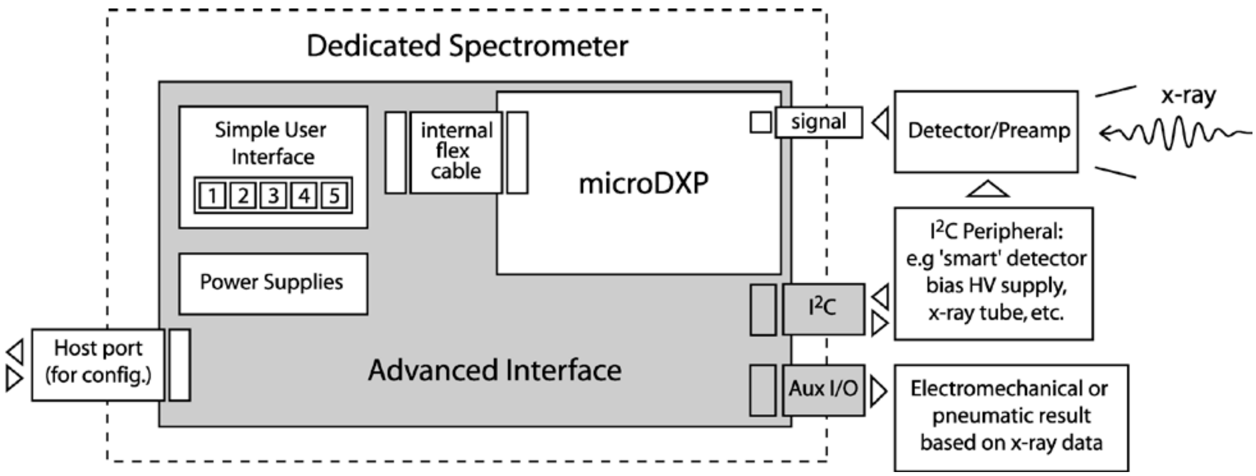

Figure 1.3: A system with a fully embedded host and user interface, with real time controls via the Auxiliary digital I/O.

Although the figure above shows a flex cable used to interface the microDXP board to the Advanced Interface board, it is also possible to use the board-to-board connector on the microDXP, as was done with the recently introduced MicroComU companion board.

Designing the system described above using conventional spectroscopy components would be a much more complex (and considerably more expensive) task, when compared with the other solutions proposed here.

1.2 Hardware Requirements

1.2.1 Host Computer or PDA

The microDXP can communicate with a host computer or PDA in a variety of ways. The hardware supports RS-232 serial communications, DSP serial communications, and parallel IDMA access to the DSP. The IDMA port allows a variety of interface options to be implemented, such as the USB 2.0 high speed interface now implemented on the MicroComU companion board. At present, DSP serial communication is not yet supported. A full description of the RS-232 command set can be found in the microDXP RS-232 Communications Specification, available online at:

http://www.xia.com/microDXP.html

1.2.2 Detector/Preamplifier

Preamplifier signal and power specifications must be verified.

The microDXP accommodates nearly all detector preamplifier signals. The two primary capacitor-discharge topologies, pulsed-reset and resistive-feedback, are supported. The voltage compliance range in the DXP analog circuitry imposes in the following constraints:

Parameter	*Minimum*	*Maximum*	*Typical*
X-ray pulse-height	250uV	375mV	25mV
Input voltage range	-	+/-5V	+/-5V

Table 1.4: Analog input signal constraints for pulsed-reset preamplifiers.

Parameter	*Minimum*	*Maximum*	*Typical*
X-ray pulse-height	250uV	625mV	100mV
Input voltage range	-	+/-5V	+/-3V
Decay time τ	100ns	infinity	50us

Table 1.5: Analog input signal constraints for resistive-feedback preamplifiers. Note that the maximum input range is less than for pulsed-reset preamplifiers.

1.2.3 Power Requirements

The microDXP requires several DC voltage supplies to operate. In cases where the user will provide their own power supply to generate these voltages, such power supplies should conform to the specifications described in section Appendix D.

The onboard digital circuitry draws from a 3.3V supply input, with on-board regulators employed to produce 2.5V for the DSP and 3.0V for the ADC. The analog circuitry runs on +/-5.0V, either supplied directly or indirectly through on-board LDO regulators.

Customers not wishing to worry about power supply requirements are advised to purchase MicroComU companion boards. The MicroComU boards will take care of generating all required voltages, at the required currents and noise performance. The MicroComU / microDXP board set may be powered from a USB port or from a single external DC supply, such as the AC wall adapter included with the USB Rapid Development Kit.

Clock Speed [MHz]	*Voltage Supply*	*Current [mA]*	*Power [mW]*	*Comment*
8	VCC	89.7	296.0	+3.3V digital – includes ADC
8	V+	20.3	101.5	+5V analog – includes VGA
8	V-	23.3	116.5	-5V analog
			514 mW	**Total power consumption at 8MHz**
16	VCC	103.0	339.9	+3.3V digital – includes ADC
16	V+	20.3	101.5	+5V analog – includes VGA
16	V-	23.3	116.5	-5V analog
			557.9 mW	**Total power consumption at 16MHz**

Table 1.6: Power consumption depends upon the pipeline clock speed.

1.2.4 Operating Environment

- Temperature Range: 0° C - 50° C
- Maximum Relative Humidity: 75%, non-condensing.
- Maximum Altitude: 3,000 meters
- Pollution degree 2
- Not rated for use in high electromagnetic fields.
- Not rated for use in environments with measurable neutron flux. Neutron flux will cause permanent damage to silicon crystals and permanently degrade or impair the performance of this system.
- The components on the microDXP board are not radiation hardened. Although there should not be a problem operating them in environments with modest gamma or X-ray flux, above a certain level this radiation will start to cause bit errors in the digital components. If necessary, please contact XIA LLC to discuss a proposed radiation environment.

1.2.5 Regulatory Compliance

The microDXP board is RoHS compliant.

1.3 Software Overview

Up to three layers of 'software' are used in a typical microDXP instrumentation system: a user interface for data acquisition and control, a driver layer that communicates between the host software and the RS-232 or USB port, and firmware code (consisting of PIC, DSP and FPGA code) that is stored and runs on the microDXP itself. Firmware code is factory pre-loaded into nonvolatile memory on the microDXP circuit board, and can be updated in the field via the XUP utility in the microManager software.

1.3.1 User Interface, e.g. microManager

The host software communicates with and directs the microDXP (via a driver layer), and displays and analyzes data as it is received. XIA offers microManager as a general-purpose configuration and data acquisition application. MicroManager features full configuration of and control over the microDXP, with intuitive data visualization, unlimited ROI's (regions of interest) Gaussian fitting algorithms and the exporting of collected spectra for additional analysis. Please refer to the **Rapid Development Kit Manual** for instructions on using microManager with the microDXP. Some users may decide instead to develop their own software to communicate with the microDXP directly via the RS-232 port. Please read §1.3.3 below for further details.

XIA offers the Accelerated DevelOPmenT (ADOPT) support option at an additional fee to such users. ADOPT is described in § 1.4.5.

1.3.2 Device Drivers, e.g. Handel

XIA will in the near future provide source code for Handel, its high-level spectrometer driver layer that operates with all DXP instruments. Handel provides an interface that is versed in lay spectroscopic units (eV, microseconds, etc...) while still allowing for safe, direct-access to the DSP. The trade-off for this ease of use is size. The Handel library is on the order of 2 Mbytes. Though microManager does currently use the Handel driver, we are in the process of refining Handel to work more efficiently with the microDXP, and thus are not yet offering Handel to microDXP users. In fact, the feature omissions in this release are due primarily to Handel, and not to the microDXP hardware or firmware. Please contact XIA for further information.

1.3.3 Physical Interface

The physical interface to the microDXP may be either an RS232 (“COM”) port or a USB 2.0 port. The MicroComU companion board is required in order to use USB communications.

Regardless of physical interface, the microDXP is controlled directly via the RS-232 command set. This is the lowest level of access currently supported and is appropriate for OEM customers who wish to design embedded systems incorporating the microDXP. Such users are encouraged to at first get up and running with microManager, and may in fact find it useful in the long run to use microManager for board configuration and their own software simply for data acquisition “in the field”. Keep in mind that certain configuration procedures, namely firmware updates via the XUP specification, can only be executed using microManager. For the complete command set please refer to the **RS-232 Command Specification**, available at:

http://www.xia.com/microDXP.html

XIA offers the Accelerated DevelOPmenT (ADOPT) support option at an additional fee to advanced users and OEM customers. ADOPT is described in §1.4.5.

1.3.4 Firmware

Firmware refers to the PIC, DSP and FiPPI (FPGA) configuration code that is stored in non-volatile memory on the microDXP. One PIC file, one DSP file, and up to three FiPPI files can be stored. For simplicity XIA provides complete firmware updates in files of the form “update_name.xup”. The XUP utility in microManager supports firmware backup and update functions. The XUP utility also supports parameters settings import and export, to aid in the production of multiple identical systems. The FiPPI and DSP are discussed in chapter 3 and chapter 5. The XUP utility is described in §1.4.1.1 and in the **Rapid Development Kit Manual**.

1.4 Support

A unique benefit of dealing with a small company like XIA is that the same people who designed them often provide the technical support for our sophisticated instruments. Our customers are thus able to get an in-depth technical advice on how to fully utilize our products within the context of their particular applications. Please read through this brief chapter before contacting us.

XIA LLC
31057 Genstar Rd.
Hayward, CA 94544 USA
(510) 401-5760

Hardware Support: microDXP@xia.com
Software Support: software_support@xia.com

Check for firmware and software updates at: http://www.xia.com/microDXP.html

1.4.1 Software and Firmware Updates

It is important that the DXP unit is being driven by the most recent software/firmware combination, since most problems are actually solved at the software level. Please check for the most up to date standard versions of the microDXP software and firmware at:

`http://www.xia.com/microDXP.html`

Please contact XIA at microDXP@xia.com if you are running semi-custom or proprietary firmware code. (*Note:* it is not a bad idea to make backup copies of your existing software and firmware before you update).

1.4.1.1 XUP Utility and Firmware Updates

Firmware updates will be provided in the XUP format, which is only supported by microManager. MicroManager's XUP utility also supports the import and export of parameter settings to and from non-volatile memory, such that multiple identical systems can easily be configured.

1.4.2 Related Documentation

As a first step in diagnosing a problem, it is sometimes helpful to consult the most recent data sheets and user manuals for a given DXP product, available in the Adobe Portable Document Format (PDF) from the XIA web site. Since these documents may have been updated since the DXP unit was purchased, they may contain information that could help solving a problem in question. All manuals, datasheets, and application notes, as well as software and firmware downloads can be found on at:

`http://www.xia.com/microDXP.html`

In order to request printed copies, please send an e-mail to software_support@xia.com, or call the company directly. In particular, we recommend that you download the following documents:

- ✓ **Rapid Development Kit User Manual** (covers both the RS232 and USB Rapid Development Kits) – All users
- ✓ **MicroComU Technical Reference Manual --** Users interested in using the MicroComU companion board to generate the microDXP's power supplies and/or communicate with the microDXP using a USB 2.0 high speed interface.
- ✓ **MicroDXP RS-232 Communications Specification** – Users who wish to develop their own software and/or hardware

1.4.3 Email and Phone Support

The microDXP comes with one year of email and phone support. Support can be renewed for a nominal fee. Please call XIA if your support agreement has expired.

The XIA Digital Processors (DGF & DXP) are digitally controlled, high performance products for X-ray and gamma-ray spectroscopy. All settings can be changed under computer control, including gains, peaking times, pileup inspection criteria, and ADC conversion gain. The hardware itself is very reliable. Most problems are not related to hardware failures, but rather to setup procedures and to parameter settings. XIA's DXP software includes several consistency checks to help select the best parameter values. However, due to large number of possible combinations the user may occasionally request parameter values which conflict among themselves. This can cause the DXP unit to report data that apparently make no sense (such as bad peak resolution or even empty spectra). Each time a problem is reported to us, we diagnose it and include necessary modifications in the new versions of our DXP control programs, as well as add the problem description to the FAQ list.

Submitting a problem report:

XIA encourages customers to report any problems encountered using any of our software. Unfortunately, due to limited resources XIA is unable to handle bug reports over the phone. In most cases, the XIA engineering team will need to review the bug information and run tests on their hardware before being able to respond.

All software-related bug reports should be emailed to software_support@xia.com and should contain the following information, which will be used by our technical support personnel to diagnose and solve the problem:

- ✓ Your name and organization
- ✓ Brief description of the application (type of detector, relevant experimental conditions...etc.)
- ✓ XIA hardware name and serial number
- ✓ Version of the library (if applicable)
- ✓ OS
- ✓ Description of the problem; steps taken to re-create the bug
- ✓ <u>Supporting data:</u>
  The most important are digital settings of the spectrometer unit, i.e., the values of the DSP parameters such as the decimation, filter length, etc.

The values of these parameters can be captured into an ASCII file in microManager. Please attach a copy of this file if possible. Capturing an oscilloscope image of the preamp output will be extremely helpful. This can done with the diagnostic tool included in microManager.

For general questions and DXP hardware issues please email: microDXP@xia.com.

1.4.4 Feedback

XIA strives to keep up with the needs of our users. Please send us your feedback regarding the functionality and usability of the microDXP and microManager software. In particular, we are considering the following development issues:

1.4.4.1 Export File Formats

We would like to directly support as many spectrum file formats as possible. If we do not yet support it, please send your specification to

`software_support@xia.com`

1.4.4.2 Fast Communications

Currently the hardware supports three communications interfaces: an RS-232 serial port, a synchronous DSP serial port and IDMA parallel DSP access. Only the RS-232 interface and recently the IDMA parallel DSP interface have been implemented thus far in software and firmware. We are interested in feedback about the support of the DSP serial port interface.

1.4.4.3 Hardware Interfaces for Production

Recently the MicroComU companion board has been introduced in order to provide customers with a USB 2.0 high speed interface to the microDXP and power supply generation for the microDXP, all in a form factor close to the size of the microDXP itself. We are interested in how well the new MicroComU board satisfies customer requirements and/or what improvements are desired.

1.4.5 The Accelerated DevelOPmenT (ADOPT) Program

The ADOPT program is a support plan for users developing custom software using any of our driver libraries. It is intended for those who wish to get direct access to the XIA software team and obtain hands-on training in the use of XIA software tools as a method of reducing overall software development time.

The standard ADOPT package provides 12 months of support divided as follows:

- 1 month: on-site support and priority phone/email support.
- 11 months: priority phone/email support.

The specific number of hours for on-site support and priority phone/email support depend on the driver library being used. Typically, the

person who will be doing the majority of the development will visit XIA for a hands-on tutorial with the XIA software team. The visitor will be encouraged to work at XIA for anywhere from a few days to two weeks, depending on the specific situation and complexity of the project. By working on-site, visitors will have access to live experimental setups on which they will be able to test their software. Furthermore, the XIA software team will be available to provide assistance and help immediately without the limitations of either email or phone.

For situations where more time is required, additional hours of support may be purchased at XIA's standard consulting rate.

This program supports both our Handel and Xerxes driver libraries as well as custom driver development. Please contact XIA to determine which driver library is right for your application (software_support@xia.com).

2 Using the microDXP

Note: *Please refer to the 'Getting Started' section of the* ***Rapid Development Kit User Manual*** *for detailed setup instructions.*

This chapter provides a general outline of microDXP operational procedures. XIA recommends using the microManager software as a microDXP *configuration* platform in all phases of production. For a step-by-step 'Getting Started' guide using microManager, including hardware setup instructions, please refer to the **Rapid Development Kit User Manual**. Though microManager also supports microDXP data acquisition (DAQ) procedures, many customers will necessarily use their own software when acquiring data. The most common procedures are explained below at the RS-232 command level. Please refer to Appendix E for a condensed summary of the RS-232 command and response protocol. Users who wish to develop the configuration routines into their software should refer to the **RS-232 Command Specification** (a separate document) for a detailed presentation of all RS-232 commands. DXP-related documents are available online at:

http://www.xia.com/microDXP.html

2.1 Board State and Configuration

The microDXP boots itself upon power up, and is shortly thereafter ready to acquire data with the same set of operating parameters used in the previous run. The first time the microDXP is powered on, detector and preamplifier related parameters should be modified (unless the default parameters have been factory-set to match the detector and preamplifier). Once these settings have been saved to non-volatile on-board memory, they will automatically load during subsequent boot operations.

2.1.1 Board Information and Status

General information about the hardware and firmware, and current board status can be retrieved.

In microManager, select **View** > **Board Info…** to display the **Board Information** dialog.

2.1.1.1 Board Information

In microManager, press **[Read Information]** to display information about the hardware and firmware configuration, variants and versions.

The RS-232 command to read board information is **0x49**. Please refer to the RS-232 Command Specification for details.

2.1.1.2 Board Status

In microManager, press **[Read Status]** to display information about the current state of the PIC, DSP and data acquisition run.

The RS-232 command to read board information is **0x4B**. Please refer to the RS-232 Command Specification for details.

2.1.2 Serial Number

In microManager, the serial number is automatically read at startup. It is displayed along the bottom of the window.

The RS-232 command to read the serial number is **0x48**. Please refer to the RS-232 Command Specification for details.

2.1.3 Firmware Version

In microManager this command is omitted. All the version information is embedded in the board information command described above.

The RS-232 command to read the firmware version information is **0x4D**. Please refer to the RS-232 Command Specification for details.

2.1.4 Idle and Sleep Modes

If the Idle Mode is enabled, the microDXP will enter a low power state a specified time after the end of a run. Several different subsystems can be manually powered down via the Sleep Mode command to conserve energy. The sleep mode can not be changed in microManager.

The RS-232 command to change the Idle Mode is **0x46**. The RS-232 command to change the Sleep Mode is **0x47**. Please refer to the RS-232 Command Specification for details.

2.1.5 On-Board Temperature

The microDXP hardware includes an I2C thermometer. The temperature reading is not accessible in microManager.

The RS-232 command to read the on-board temperature is **0x41**. Please refer to the RS-232 Command Specification for details.

2.2 Global Settings and the GLOBSET

The GLOBSET, specified in Appendix A, contains global settings including detector/preamplifier settings and system settings. There is only one GLOBSET—these settings are used for all peaking times and MCA formats.

The GLOBSET includes detector/preamplifier settings, advanced processor settings, run control settings and diagnostic control settings.

In microManager, GLOBSET settings are accessed via the **Detector** and **Advanced** tabs of the **Settings** panel. Because these settings are global, changes are simultaneously applied and saved to nonvolatile memory via the **[Apply And Save]** button.

2.2.1 Preamplifier Settings

The microDXP's analog signal conditioner must be configured for the specific detector / preamplifier that is used in order for the downstream digital processing pipeline to operate correctly. The microDXP *must* have firmware pre-loaded in non-volatile memory that is appropriate for the preamplifier type used (i.e. reset-type or RC-feedback), and parameters must be properly set based upon the following information, which is normally included in the detector or preamplifier documentation (or alternatively measured using an oscilloscope).

- ✓ The decay constant for RC-type preamplifiers, or the settling time of the reset transient for reset-type preamplifiers.
- ✓ The preamplifier signal polarity: Positive polarity means that an x-ray produces a positive step in voltage at the preamplifier output. The preamplifier's signal polarity unrelated to the detector's bias voltage polarity.

Note: *The microDXP must have appropriate firmware for the* ***preamplifier type*** *used. In addition the* ***HARDWARE*** *must also be set to the appropriate preamplifier type via a miniature DIP switch* ***S1****.* ***Set S1 to "RAMP" for reset-type preamplifiers. Set S1 to "OFFSET" for RC-type preamplifiers.***

2.2.1.1 Preamplifier Type

Please refer to § 4.1 for more detailed description of charge-sensitive preamplifier topologies. Briefly, RC-feedback preamplifiers use resistive feedback to provide a continuous discharge path for the feedback capacitor, resulting in a characteristic decay time, e.g. 50μs. Reset preamplifiers employ a switch to periodically discharge the capacitor very quickly, resulting in a periodic 'staircase' waveform with many x-ray steps between each reset.

The microDXP *must* have DSP *and* FPGA code pre-loaded in non-volatile memory that is appropriate for the preamplifier type used. In addition the microDXP hardware must be set *type via a miniature DIP switch* (refer to Appendix D for the switch location):

Set to 'RAMP' for reset-type preamplifiers.
Set to 'OFFSET' for RC-feedback preamplifiers.

In MicroManager, the preamplifier type corresponding to the loaded firmware is displayed in the **Detector** tab of the **Settings** panel.

2.2.1.2 Decay Time: TAURC

If using an RC-feedback type preamplifier you must set the DSP parameter TAURC, where the decay time constant τ_{RC}, is expressed in microseconds [μs]:

$$\mathrm{TAURC} = \tau_{RC}$$

In microManager, TAURC is set via the **Decay Time** field in the **Detector** tab of the **Settings** panel. Simply enter the desired value in microsecond units and press the **[Apply And Save]** button.

The RS-232 command to set and save TAURC to nonvolatile memory is **0x89**. Please refer to the RS-232 Command Specification for details.

2.2.1.3 Reset Interval: RESETINT

If using a reset-type preamplifier you must set the DSP parameter RESETINT to the reset delay time or reset interval. This is the period after each preamplifier reset that the microDXP waits before re-enabling data acquisition.

RESETINT, expressed in microseconds, is set based on the settling time of the preamplifier reset transient waveform, which typically ranges from hundreds of nanoseconds to hundreds of microseconds. Setting the delay shorter than the transient settling time typically degrades the energy resolution and may even introduce ‘reset artifact’ events into the spectrum. Setting the delay longer than necessary introduces additional processor dead time, which will reduce the data throughput at high count rates.

In microManager, RESETINT is set via the **Reset Interval** field in the **Detector** tab of the **Settings** panel. Simply enter the desired value in microsecond units and press the **[Apply And Save]** button.

The RS-232 command to set and save RESETINT to nonvolatile memory is **0x8A**. Please refer to the RS-232 Command Specification for details.

2.2.1.4 Pre-Amplifier Signal Polarity

Preamplifier signal polarity denotes the polarity of the preamplifier output signal. Please read through § 4.1 for a description and figures relating to the preamplifier signal polarity. Briefly, a positive polarity preamplifier produces a voltage step with a rising edge. The DSP parameter POLARITY must be set correctly:

POLARITY = 0, for negative polarity

POLARITY = 1, for positive polarity

In microManager, POLARITY is set via the **Polarity** field in the **Detector** tab of the **Settings** panel. Select the desired polarity and press the **[Apply And Save]** button.

The RS-232 command to set and save POLARITY to nonvolatile memory is **0x87**. Please refer to the RS-232 Command Specification for details.

2.2.2 Advanced Processor Settings

The advanced processor settings enable and disable various Digital X-ray Processor functions, and correspond to bits of the DSP parameter RUNTASKS. Typically these settings should only be modified as directed by XIA LLC engineers.

In microManager, the advanced processor settings are accessed in the **Advanced** tab of the **Settings** panel. Please refer to Chapter 5 and specifically section 5.8 for further details.

2.3 MCA Settings and GENSETs

The GENSET, specified in Appendix B, is a table of MCA-related parameters, e.g. the number of bins and bin granularity, preset length of run, etc. Parameters within the GENSET can be modified and stored such that a standardized MCA format can be implemented with a single command. Five (5) GENSETs, and thus five MCA formats, can be stored and retrieved.

2.3.1 Selecting the GENSET

In microManager, GENSET settings are accessed via the **Acquisition** tab of the **Settings** panel. GENSETs 0-4 can be selected via the **GENSET** drop-down list. Modifications to MCA settings can be tested by pressing the **[Apply]** button, and saved to nonvolatile memory via the **[Save]** button.

The RS-232 command to select a GENSET is **0x83**. Note that this command simply retrieves one of five tables of MCA settings from nonvolatile memory. Please refer to the RS-232 Command Specification for details.

2.3.2 MCA Size

This command is used to change the number of bins in the multi-channel analyzer (MCA) via the DSP parameter MCALEN.

In microManager, MCALEN is set via the **Number of MCA Bins** field in the **Acquisition** tab of the **Settings** panel. Select the desired number and press the **[Apply]** button to test the setting. Press the **[Save]** button to store the setting to the currently selected GLOBSET in nonvolatile memory.

The RS-232 command to set MCALEN is **0x85**. Note that this command does not save the setting to nonvolatile memory. Please refer to the RS-232 Command Specification for details.

2.3.3 MCA Granularity (Bin Width)

The granularity setting determines the hardware (DSP) scaling factor BINGRANULAR, with four standard settings and a custom as outlined in table **Error! Reference source not found.** below. The raw energy filter sample (i.e. the 'energy') is divided in the DSP by BINMULTIPLE. The combination of MCA size and granularity must satisfy dynamic range constraints imposed by the digital filter pipeline. If they do not, a *dead* region will be included at the highest energies in the spectrum, i.e. there is no possibility of getting counts in this region.

Recommended # of MCA Bins	*Absolute MAX # of MCA Bins*	*BINGRANULAR*	*BINMULTIPLE*
4096	8192	0	1
2048	8192	1	2
1024	4096	2	4
512	2048	3	8
< 512 (e.g. 128)	-	4	(**Bin Size**) (e.g. 32)

Table 2.1: Suggested **Bin Granularity** settings based upon the **Number of MCA Bins**. Using a larger-than-recommended number of bins for a given granularity will result in extended-range events—if such x-rays are present—being displayed. If greater than the *absolute maximum*, the spectrum will include a high-energy dead region.

In microManager, BINGRANULAR is set via the **Bin Granularity** field in the **Acquisition** tab of the **Settings** panel. Select the desired setting and press the **[Apply]** button to test. Note that the **Bin Size** changes as a result. Press the

[Save] button to store the setting to the currently selected GLOBSET in nonvolatile memory.

The RS-232 command to set BINGRANULAR is **0x84**. Note that this command does not save the setting to nonvolatile memory. Please refer to the RS-232 Command Specification for details.

2.3.4 Base Gain

To maintain perfect energy calibration a unique analog gain setting can be stored for every combination of peaking time and MCA format. The **Base Gain** is the approximate gain value appropriate for a given MCA size and dynamic range. The scale is 0 to 100; the value 100 applies maximum gain to the input signal.

In microManager, we recommend using the ROI calibration routine described in the Rapid Development Kit User Manual to set the **Base Gain. Base Gain** can also be edited directly in the **Acquisition** tab of the **Settings** panel. Select the desired setting and press the **[Apply]** button to test. Press the **[Save]** button to store the setting to the currently selected GLOBSET in nonvolatile memory.

The RS-232 command to set the base GAINDAC value is **0x88**. Note that this command does not save the setting to nonvolatile memory. Please refer to the RS-232 Command Specification for details.

2.3.5 Reading the Current GENSET

The RS-232 command to read the current GENSET table is **0x8E**. Note that this command does not save the setting to nonvolatile memory. Please refer to the RS-232 Command Specification for details.

2.3.6 Saving the Current GENSET to Non-Volatile Memory

The RS-232 command to save the current GENSET table is **0x8F**. Please refer to the RS-232 Command Specification for details.

2.4 Spectrometer Settings and PARSETs

For a given FiPPI decimation (i.e. peaking time *range*) and pipeline clock speed, the slow filter length SLOWLEN sets the peaking time. In practice other FiPPI control parameters, as well as other DSP parameters such as the analog gain, may also be require adjustment for optimal performance at each peaking time. To avoid repeating the optimization procedure five sets of parameters, and thus five optimized peaking times are available for each FiPPI decimation.

The PARSET, specified in Appendix C, is a table of peaking-time-related spectrometer parameters, e.g. filter values, thresholds, pileup inspection settings, etc. Parameters within the PARSET can be modified and stored such that a calibrated spectrum is achieved whenever the peaking time (PARSET) is subsequently selected. The PARSETs are stored in the nonvolatile flash memory. Five (5) PARSETs are available for each FiPPI decimation purchased

(see §1.1.2.4). Up to three FiPPI decimations, and thus a maximum of 15 optimized peaking times can be stored.

The PARSET further contains 5 sets of gain tweaking and threshold settings, each corresponding to the 5 GENSETs or MCA formats. Storing thresholds and gain settings for every combination of peaking time and MCA format eliminates the need for calibrating a given combination more than once.

The factory-set default spectrometer settings should be adequate to acquire a recognizable spectrum. To achieve optimal performance the spectrometer settings must be adjusted, and stored to non-volatile memory such that the optimized settings will be accessible in the future. To maximize throughput, the slow filter peaking time τ_{PS} should be chosen to be as short as possible to meet energy resolution requirements, since the maximum throughput scales as $1/\tau_{PS}$ (see §4.10 for a detailed discussion of throughput):

$$OCR_{max} = 1/(e\ \tau_d) = 0.37/\tau_d$$

2.4.1 Selecting a FiPPI Decimation

The microDXP comes pre-programmed with at least one and up to three FiPPI decimations or peaking time ranges.

In microManager, the FiPPI decimation is accessed via the **Acquisition** tab of the **Settings** panel. FiPPIs 0-2 can be selected via the **Peaking Time Range** drop-down list. Notice that the **Peaking Time** and other PARSET settings update when the new FiPPI decimation is selected.

The RS-232 command to select a FiPPI decimation is **0x81**. Please refer to the RS-232 Command Specification for details.

2.4.2 Selecting a PARSET

For each peaking time range, five (5) tables of parameters are stored in nonvolatile memory. Each PARSET corresponds to a peaking time.

In microManager, PARSET settings are accessed via the **Acquisition** tab of the **Settings** panel. PARSETs 0-4 can be selected via the **Peaking Time** drop-down list. Modifications to spectrometer settings for the selected PARSET can be tested by pressing the **[Apply]** button, and saved to nonvolatile memory via the **[Save]** button.

The RS-232 command to select a PARSET is **0x82**. Note that this command simply retrieves one of five tables of spectrometer settings from nonvolatile memory. Please refer to the RS-232 Command Specification for details.

2.4.3 Filter Parameters

Although the peaking time itself cannot be edited, several digital filter parameters are available for modification. Please refer to Chapter 4 for a thorough discussion of digital filtering with the DXP.

In microManager, the filter parameters are accessed via the **Acquisition** tab of the **Settings** panel. Press the **[Edit Filter Parameters]** button and modify settings in the associated dialog. Modifications for the selected PARSET can be

tested by pressing the **[OK]** button, and saved to nonvolatile memory via the **[Save]** button.

The RS-232 command to modify filter parameters is **0x8B**. Note that this command does not save the change to nonvolatile memory. To do so it is necessary to subsequently save the current PARSET as described below. Please refer to the RS-232 Command Specification for details

2.4.4 Baseline Average Length

A running average of baseline measurements is computed, which is then subtracted from sampled peak values to compute the energy of corresponding incident x-rays. The number of baseline samples averaged is set in microManager as **Baseline Average Length**. In the DSP this is converted into the parameter BLFILTER according to the equation:

Baseline Average Length = 32768 / BLFILTER

Please refer to section 4.4 for a thorough discussion of baseline averaging.

In microManager, BLFILTER is accessed via the **Acquisition** tab of the **Settings** panel. Select the desired **Baseline Average Length** from the drop-down list. Modifications for the selected PARSET can be tested by pressing the **[Apply]** button, and saved to nonvolatile memory via the **[Save]** button.

The RS-232 command to modify BLFILTER is **0x92**. Note that this command does not save the change to nonvolatile memory. To do so it is necessary to subsequently save the current PARSET as described below. Please refer to the RS-232 Command Specification for details.

2.4.5 Thresholds

Proper triggering on input events depends on good threshold settings, particularly for the so-called **Trigger** (fast filter) and **Baseline** (intermediate filter) thresholds. The DSP parameters THRESHOLD, BASETHRESH and SLOWTHRESH correspond to thresholds applied to the **Trigger** (fast), **Baseline** (intermediate), and **Energy** (slow) filters, respectively. Please refer to section 5.10.4 for a thorough discussion of thresholds.

Each PARSET includes 5 different settings for THRESHOLD, BASETHRESH and SLOWTHRESH, corresponding to the 5 MCA formats, or GENSETs.

In microManager, the threshold values for the current PARSET and GENSET are accessed via the **Acquisition** tab of the **Settings** panel. Enter the desired threshold settings. Settings for the current PARSET can be tested by pressing the **[Apply]** button, and saved to nonvolatile memory via the **[Save]** button.

The RS-232 command to modify thresholds is **0x86**. Note that this command does not save the change to nonvolatile memory. To do so it is necessary to subsequently save the current PARSET as described below. Please refer to the RS-232 Command Specification for details.

2.4.6 Fine Gain Trim

To maintain calibration across all combinations of peaking times and MCA formats, slight modifications to the analog gain are necessary. The DSP PARSET parameter GAINTWEAK is combined with GENSET parameter GAINBASE to arrive at the final gain setting. Each PARSET includes 5 different settings for GAINTWEAK, corresponding to the 5 MCA formats, or GENSETs.

In microManager, the threshold values for the current PARSET and GENSET are accessed via the **Acquisition** tab of the **Settings** panel. Enter the desired threshold settings. Settings for the current PARSET can be tested by pressing the **[Apply]** button, and saved to nonvolatile memory via the **[Save]** button.

The RS-232 command to modify thresholds is **0x86**. Note that this command does not save the change to nonvolatile memory. To do so it is necessary to subsequently save the current PARSET as described below. Please refer to the RS-232 Command Specification for details.

2.4.7 Reading the Current PARSET

The RS-232 command to read the current PARSET table is **0x8C**. Note that this command does not save the setting to nonvolatile memory. Please refer to the RS-232 Command Specification for details.

2.4.8 Saving the Current PARSET to Non-Volatile Memory

In microManager, all changes to the current PARSET can be saved to nonvolatile memory via the **[Save]** button.

The RS-232 command to save the current PARSET table is **0x8D**. Please refer to the RS-232 Command Specification for details.

2.5 Repetitive Configuration of Identical Systems

In cases where many microDXPs are to be configured identically (or nearly identical) it is very desirable to 'carbon-copy' user settings that have been optimized. This procedure is supported in microManager. Note that gain variations between modules make fine gain tuning necessary for each microDXP.

2.5.1 Create Master Parameter Set…

In microManager select **Firmware > Create Master Parameter Set…** to save all parameters to XUP file format. Select the desired filename and location and press **[Save]**.

2.5.2 Download a Master Parameter Set…

In microManager select **Firmware > Download…** to write the parameters in the XUP file you created to a second microDXP. Browse to the desired XUP file and location and press **[Download]**.

2.6 Data Acquisition:

This section describes the most common data acquisition procedures. The most common commands (i.e. start/stop run, readout data) are covered in §**Error! Reference source not found.**. Please refer to the **RS-232 Command Specification** (a separate document) for a detailed presentation of all RS-232 commands. DXP-related documents are available online at:

http://www.xia.com/microDXP.html

In microManager select the **MCA** tab in the **Data Acquisition** panel.

2.6.1 Starting a Run

Data acquisition runs can be configured to automatically terminate the run after a preset time or number of input or output counts has elapsed, as described below in §2.6.5. By default (i.e. PRESET = 0) the run continues until a 'stop run' command is issued (see §2.6.2).

In microManager, simply press the **[Start Run]** button.

The RS-232 command to start a data acquisition run is **0x00**. Please refer to the RS-232 Command Specification for details.

2.6.2 Stopping a Run

By default (i.e. PRESET = 0) a run in progress continues until a 'stop run' command is issued. The microDXP can be configured to automatically terminate until a preset time or number of input or output counts has elapsed, as described below in §2.6.5. In such cases the 'stop run' command overrides the run preset.

In microManager, simply press the **[Stop Run]** button.

The RS-232 command to stop a data acquisition run is **0x01**. Please refer to the RS-232 Command Specification for details.

2.6.3 Reading a Spectrum

The 'read spectrum' command supports readout of any contiguous region of the MCA data, extending, of course to the entire spectrum. Each MCA bin is represented in DSP program memory as a 24-bit word (i.e. 3 bytes), supporting up to 16,777,215 counts per bin. In some cases (i.e. for short runs and/or low count rates) the upper bits of each bin word will always be zero. The readout speed can be increased by opting to read out a fewer number of bytes per bin, or DEPTH. At DEPTH=2 up to 65,535 counts per bin are supported. At DEPTH=1 up to 255 counts per bin are supported. *Note:* If the number of counts in a bin exceeds the DEPTH, the resulting distribution will display sharp discontinuities. The data is however *always* stored internally in the DSP at the full 24 bits. It is thus not necessary to restart the run when the DEPTH is exceeded: Simply change DEPTH and re-read the spectrum.

In microManager, the spectrum and statistics can be updated automatically or manually. Check the **Continuous?** checkbox for automatic updates. If the

checkbox is unchecked press the **[Update]** button to manually update the spectrum and statistics.

The RS-232 command to read the MCA spectrum is **0x02**. Please refer to the RS-232 Command Specification for details.

2.6.4 Reading (and Calculating) the Run Statistics

The ‘read run statistics’ command retrieves the fast-filter livetime (LIVETIME), the realtime (REALTIME), the number of input counts (FASTPKS) and the number of output counts (EVENTSINRUN). These parameters can be used to directly calculate the input count rate (ICR), output count rate (OCR) and the deadtime percentage ($\%_{DEADTIME}$):

$$ICR = FASTPKS/LIVETIME$$

$$OCR = EVENTSINRUN/REALTIME$$

$$\%_{DEADTIME} = OCR/ICR$$

Realtime and livetime are expressed in units of 500 ns. *Note:* The LIVETIME corresponds to the fast filter—NOT the energy filter—and thus does not alone determine the relationship between input and output count rates, i.e. the deadtime percentage.

In microManager, the spectrum and statistics can be updated automatically or manually. Check the **Continuous?** checkbox for automatic updates. If the checkbox is unchecked press the **[Update]** button to manually update the spectrum and statistics.

The RS-232 command to read the run statistics is **0x06**. Please refer to the RS-232 Command Specification for details.

2.6.5 Specifying fixed run lengths

By default, the microDXP acquires data until a stop command is received from the host. Alternatively the microDXP can automatically terminate data acquisition runs based upon the realtime, livetime or the number of input or output counts exceeding a preset value. *Note:* Although the realtime and livetime are expressed in (measured accurately to) units of 500 nanoseconds, the process that monitors the realtime and livetime is only updated every 500 microseconds. Similarly, the input and output counts are tallied every 500 microseconds. The result is that a preset of 100,000 input counts may terminate with a slightly larger number of input events than 100,000. Nonetheless, the run statistics (see §2.6.4 above) are all mutually consistent and accurate.

In microManager, select the desired **Preset Run Type** from the drop-down list, and enter the desired value in the **Preset Value** field.

The RS-232 command to control preset run settings is **0x07**. Please refer to the RS-232 Command Specification for details.

2.7 Diagnostic Tools

The microDXP provides for diagnostic features including ADC trace readout, baseline trace and histogram readout and DSP parameters readout.

2.7.1 ADC Trace Readout

An 8000 point digital trace of the signal at the ADC is available for readout as a diagnostic aid. Remember that this signal is NOT the raw preamplifier signal, but has been conditioned as described in section 3.2.

In microManager, select the **ADC** tab in the **Data Acquisition** panel. Enter the desired value in the **Sampling Interval** field. The minimum value is 0.125 microseconds. Press the **[Read ADC]** button to refresh the display. Press the **[Save ADC]** to save the data to file in ASCII format.

The RS-232 command to read the ADC is **0x11**. Please refer to the RS-232 Command Specification for details.

2.7.2 Baseline Diagnostics

An 8000 point digital trace of the running baseline average is available for readout as a diagnostic aid. A histogram of instantaneous baseline samples is also available. Refer to sections 4.4 and 5.5 for discussion of baseline acquisition and averaging.

In microManager, select the **Baseline** tab in the **Data Acquisition** panel. Press the **[Get Baseline]** button to display the baseline histogram. Press the **[Get Baseline History]** button to display the baseline average vs. time. Press the **[Save Baseline]** to save the data to file in ASCII format.

The RS-232 command to read the baseline histogram is **0x10**. The RS-232 command to read the baseline history is **0x12**. Please refer to the RS-232 Command Specification for details.

2.7.3 DSP Parameters Readout

In microManager, the spectrum and statistics can be updated automatically or manually. Check the **Continuous?** checkbox for automatic updates. If the checkbox is unchecked press the **[Update]** button to manually update the spectrum and statistics.

The RS-232 command to read the run statistics is **0x06**. Please refer to the RS-232 Command Specification for details.

3 MicroDXP Functional Description

3.1 Organizational Overview

The DXP channel architecture is shown in Figure 3.1. The four major operating blocks are the Analog Signal Conditioner (ASC), Digital Filter, Peak Detector, and Pileup Inspector (FiPPI), Digital Signal Processor (DSP), and PIC microcontroller. Also depicted are the ADC, the two host interface connections, a digital temperature sensor and the nonvolatile memory. The functions of each block are summarized below. This chapter does assume the reader has some familiarity with x-ray pulse processing theory and electronic devices. Please see Chapter 4 of this manual for a brief review.

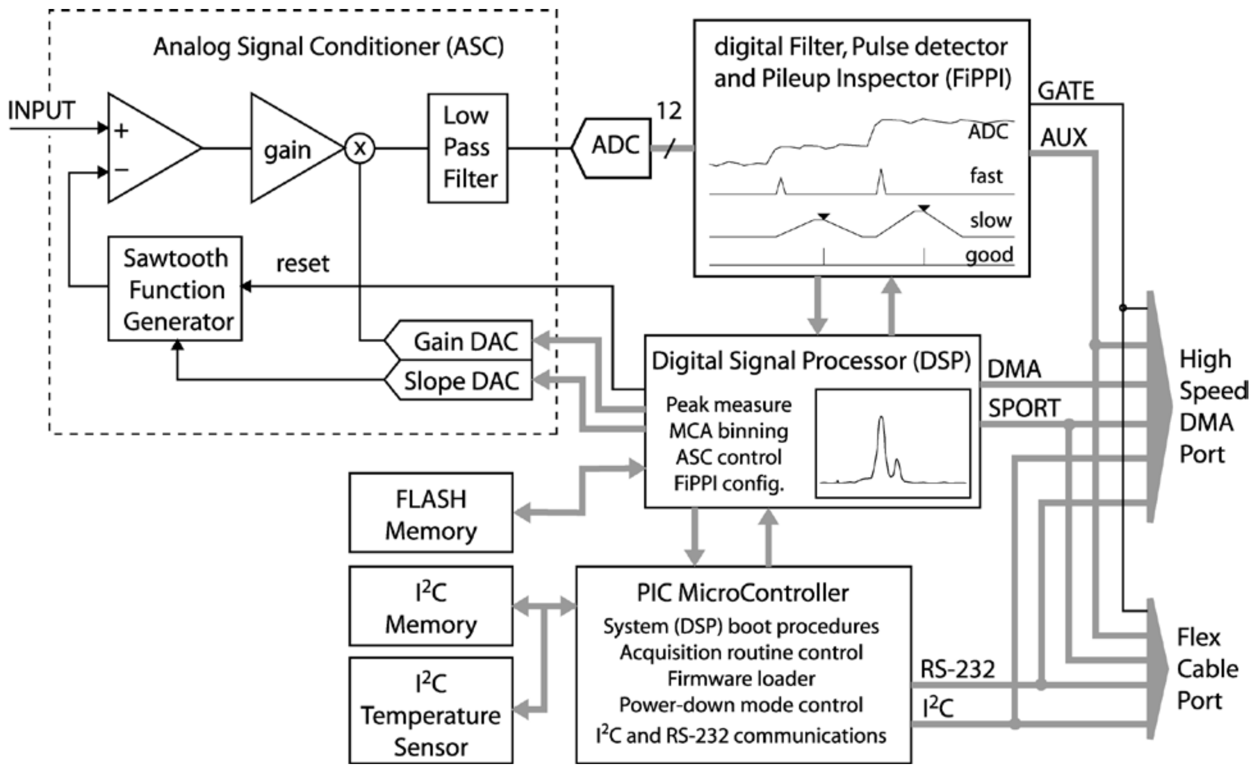

Figure 3.1: Block diagram of the DXP channel architecture, showing the major functional sections and interface port options.

3.2 The Analog Signal Conditioner (ASC)

The ASC has two major functions: to reduce the dynamic range of the input signal so that it can be adequately digitized by a 12 bit converter and to

reduce the bandwidth of the resultant signal to meet the Nyquist criterion based upon the ADC sample rate.

3.2.1 Dynamic Range Reduction

In many cases, and particularly for reset-type preamplifiers, the full-scale output voltage range is many times greater than the voltage step produced by a single x-ray event (see Figure 4.2). A high sampling rate is necessary to provide good pulse pileup detection, as described in §4.8, and sufficient ADC resolution is required to accurately sample the noise prior to the digital filters. For high count rates, pulse pair resolution less than 200 ns is desirable, which implies a sampling rate of 10 MSA or more. In order to reduce the noise σ in measuring V_x (see Figure 4.1 and Figure 4.3), experience shows that σ must be at least 4 times the ADC's single bit resolution ΔV_1. This effectively sets the gain of the amplifier stages preceding the ADC. Then, if the preamplifier's full scale voltage range is V_{max}, it must digitize to N bits, where N is given by:

$$N = \log_{10} (V_{max}/\Delta V_1)/\log_{10} (2)$$

Equation 3-1

For a typical high-resolution spectrometer, N must at least 14. An ADC with 14 *effective* bits resolution that supports 10 MSA is quite expensive. The alternative approach is to first reduce the dynamic range of the preamplifier output signal such that a moderately priced ADC can be used.

3.2.1.1 Reset-Type Preamplifiers

For reset-type preamplifiers the dynamic range reduction is accomplished using a novel dynamic range technology, for which XIA has received a patent, and which is indicated in Figure 3.2. Here a preamplifier output is shown which cycles between about -3.0 and -0.5 volts. We observe that it is not the overall function which is of interest, but rather the individual steps, such as shown in Figure 4.2 of the next chapter, that carry the x-ray amplitude information. Thus, if we know the average slope of the preamp output, we can generate a saw tooth function that has this average slope and restarts each time the preamplifier is reset, as shown in Figure 3.2. If we then subtract this saw tooth from the preamplifier signal, we can amplify the difference signal to match the ADC's input range, also as indicated in the Figure.

The generator required to produce this saw tooth function is quite simple, comprising a current integrator with an adjustable slope and a reset switch. A DAC (SLOPEDAC) controls the current, which sets the slope. The DAC input value is set by the DSP, which thereby gains the power to adjust the saw tooth generator in order to maintain the ASC output (i.e. the "Amplified Saw tooth Subtracted Data" of Figure 3.2) within the ASC input range.

In practice, the large-signal dynamic range can be reduced by a factor of 8 to 16, thus reducing the required number of bits necessary to achieve the same resolution from 14 to 10.

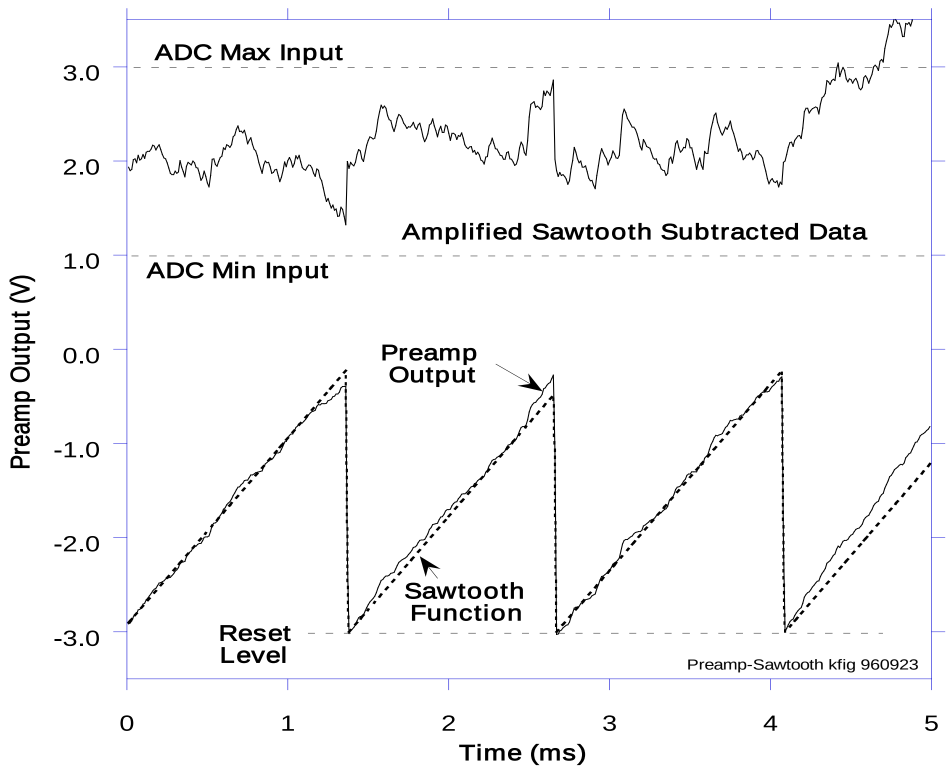

Figure 3.2: A saw tooth function having the same average slope as the preamp output is subtracted from it and the difference amplified and offset to match the input range of the ADC.

Occasionally, as also shown in Figure 3.2, fluctuations in data arrival rate will cause the conditioned signal to pass outside the ADC input range. This condition is detected by the FiPPI, which has digital discrimination levels set to ADC zero and full scale, which then interrupts the DSP, demanding ASC attention. The DSP remedies the situation by quickly closing the reset switch. During this time, data passed to the FiPPI are invalid. Preamplifier resets are detected similarly. When detected the DSP responded by resetting the current integrator until the signal comes back into range. *Note:* Data acquisition is halted until the time period defined by RESETINT has expired (see also §2.2.1.3).

3.2.1.2 RC-Feedback Preamplifiers

RC-feedback preamplifiers inherently produce a smaller dynamic range: At low rates the preamplifier output decays to baseline after each event, producing a voltage range on the order of a single event; at higher rates successive events add up, however, the larger the sum, the steeper the decay slope back to baseline. The result, as of course intended, is to yield a full-scale signal that increases only logarithmically with count rate.

Another consideration with RC-feedback preamplifiers is the DC offset. Although many such preamplifiers produce a ground-referenced output, many do not. Both problems are solved in the microDXP by adding a digitally controlled DC offset voltage to the input signal. The offset DAC is used both to

subtract the DC shift that results when running at high rates, and to account for any DC offset voltage in the preamplifier output signal.

3.2.2 Nyquist Criterion

The Nyquist criterion states that there should be no frequency component in the signal that exceeds half of the sampling frequency. Frequencies above this value are aliased into the digitized signal at where they are indistinguishable from original components at those frequencies. In particular, high frequency noise would appear as excess low frequency noise, spoiling the spectrometer's energy resolution.

3.3 The Analog to Digital Converter (ADC)

Signal digitization occurs in the Analog-to-Digital converter (ADC), which lies between the ASC and the FiPPI. The ADC is a 12 bit device, which is currently run at either 8 or 16MHz. Although the chosen ADC supports both sampling rates, because the Nyquist criterion must be satisfied in both cases (i.e. 4MHz at 8MSA and 8MHz at 16MSA) the passive components are different in each case. Changing the pipeline (i.e. ADC and digital filtering) clock speed thus requires modification to the microDXP hardware.

Pipeline Clock Speed	*ADC Sampling Rate*	*Nyquist Frequency*	*Sample Period*
8 MHz	8 MSA	4 MHz	125 ns
16 MHz	16 MSA	8 MHz	62.5 ns

Figure 3.3: Nyquist frequencies and sample periods for the two pipeline clock speed offerings.

3.4 The Filter, Pulse Detector, & Pile-up Inspector (FiPPI)

The FiPPI is implemented in a field programmable gate array (FPGA) to accomplish the various filtering, pulse detection and pileup inspection tasks discussed in chapter 4. As described there, it utilizes up to three digital filters running simultaneously for the purposes of pulse detection, pileup inspection and rejection, noise reduction and peak selection.

The FiPPI also includes a livetime counter that is activated any time the DSP is enabled to collect x-ray pulse values from the FiPPI and therefore provides an extremely accurate measure of the system livetime. In particular, as described in §3, the DSP is not live, while an x-ray pulse is being processed, during preamplifier resets or during ASC out-of-ranges; in the latter case both because it is adjusting the ASC and because the ADC inputs to the FiPPI are invalid. Thus the DXP measures livetime more accurately than an external clock, which is insensitive to resets and includes them as part of the total livetime. While the average number of resets/sec scales linearly with the count rate, in any given measurement period there will be fluctuations in the number of resets which may affect counting statistics in the most precise measurements.

3.4.1 FiPPI Decimation

FiPPI's are distinguished also by 'decimation'. Decimation refers to pre-averaging of the ADC signal prior to the FPGA processing pipeline. Each decimation accommodates a specific range of peaking times, i.e. shaping or integration times. Up to three (3) FiPPI configuration files can be stored in the microDXP's nonvolatile memory. When the peaking time is changed such that a range boundary is crossed, the host software downloads the appropriate FiPPI configuration to the MicroDXP.

FiPPI Decimation	*#ADC Samples in Average*	*Peaking Time Range:* ***16MHz Pipeline Clock***	*Peaking Time Range:* ***8MHz Pipeline Clock***
0	1	**125 ns** – 750 ns	250 ns – 1.5 µs
1	2	250 ns – 1.5 µs	500 ns – 3.0 µs
2	4	500 ns – 3.0 µs	1.0 µs – 6.0 µs
3	8	1.0 µs – 6.0 µs	2.0 µs – 12.0 µs
4	16	2.0 µs – 12.0 µs	4.0 µs – 24.0 µs
5	32	4.0 µs – 24.0 µs	8.0 µs – 48.0 µs
6	64	8.0 µs – 48.0 µs	16.0 µs – **96.0 µs**

Table 3.1: Available peaking time ranges by FiPPI decimation and pipeline clock speed.

3.4.2 FiPPI Code Variants

The FiPPI pipeline topology for RC-type preamplifiers is different than for reset-type preamplifiers, thus two standard code variants are offered for each decimation. Additionally, any use of the auxiliary digital I/O will require per-instance FiPPI configuration variant. Please contact XIA to discuss this development.

3.5 The Digital Signal Processor (DSP)

The Digital Signal Processor acquires and processes event data from the FiPPI, and controls the ASC through DACs. The processor is an Analog Devices ADSP-2183 16 bit Fixed-Point DSP optimized for fixed-point arithmetic and high I/O rates. Different DSP program variants are used for different types of data acquisition and different preamplifier types. Chapter 5 describes in detail the DSP operation, its tasks, and parameters which control them.

The ADSP-2183 has 16K words of 16-bit wide data memory and 16K words of 24-bit wide program memory, part of which is used as data memory to hold the MCA spectrum. (If more memory is required for special purposes, up to 4 Mbytes of extended memory can be added by specifying option M). Transferring data to/from these memory spaces is done through the DSP's built-in DMA port, which does not interfere with the DSP program operation.

3.5.1 FLASH Memory

A new feature implemented on the microDXP is the inclusion of on board non-volatile memory, which allows for firmware storage and retrieval.

The flash memory, accessed by the DSP, used to store FiPPI configuration codes and parameter sets, called GENSETS and PARSETS. The FiPPI is thus configured and optimized independently, with only a short command issued from the host.

Parameter sets simplify data acquisition procedures. The DXP works well only if the internal parameters that govern digital filtering, peak detection and pileup inspection are properly set. For the lay user the optimization process can become overwhelming. The new approach is to optimize the relevant parameters for each peaking time only one time, and store the entire parameter in a unique location in the flash memory. The exact state can be subsequently reproduced simply by selecting the saved parameter set. The flash memory can be updated with new FiPPI code via the RS-232, SPORT or IDMA ports.

3.5.2 Serial Port (SPORT)

The Analog Devices DSP synchronous serial port, or SPORT, supports a variety of serial data communications protocols, and offers a maximum transfer rate of approximately 1Mbyte/sec. The SPORT interface is a interface candidate in multi-microDXP systems, and is available with either interface connector option (see §3.7 below),

3.5.3 DMA Port

Parallel Direct Memory Access (DMA) provides the highest bandwidth communications path to the DSP data memory. Transfer rates up to 16 Mbytes/sec are possible. The DMA bus is available to the host computer/PDA only if the high-speed board-to-board interface option (see §3.7.2 below) is chosen.

3.5.4 DSP Code Variants

The FiPPI pipeline topology for RC-type preamplifiers is different than for reset-type preamplifiers, thus two standard code variants are offered for each FiPPI decimation. Additionally, special data acquisition modes (e.g. time-resolved spectroscopy, multi-SCA's, etc.) require variation in the DSP code. Please contact XIA to discuss this development.

3.6 PIC MicroController

The PIC microprocessor serves as the system controller, carrying out procedures to boot the board, loading appropriate DSP code from memory, and running acquisition routines. In addition the PIC handles I/O including RS-232 standard communications and an I^2C bus for controlling dedicated peripheral devices.

3.6.1 RS-232 Serial Port

The RS-232 serial port is the default communications interface for the microDXP, and is wired to both the flex-cable connector and the high-speed DMA port connector. Though relatively slow (115 kbaud) the RS-232 port is in fact adequate for most applications. As one of the oldest communications

standards, RS-232 enjoys wide compatibility with existing devices and is supported by all x86 Personal Computers.

3.6.2 I^{2}C Serial Bus

The I^{2}C serial protocol allows for several serial devices to share the same two-wire bus through a device ID handshaking procedure. I^{2}C devices are pre-programmed with a four-bit device ID (e.g. 1001 is used for some digital temperature sensors) appended with a 3-bit suffix that is typically set by hardwiring the appropriate pins to either the supply voltage or to ground. Two I^{2}C devices are included on the microDXP itself (described below), and the bus is wired to both interface connectors to provide for microDXP control over, or monitoring of, other devices.

3.6.3 I^{2}C Memory

A new feature implemented on the microDXP is the inclusion of on board non-volatile memory, which allows for firmware storage and retrieval. The I^{2}C memory, accessed by the PIC, is used to store the DSP code and general system information. The DSP is booted automatically upon power-up. The I^{2}C memory can be updated with new DSP code via the RS-232 serial port.

3.6.4 I^{2}C Temperature Sensor

An I^{2}C temperature sensor is included on the microDXP. The temperature measurement range is –55°C to +125°C, with ± 1°C accuracy.

3.6.5 PIC Code Variants

Semi-custom and custom PIC code will be necessary for applications utilizing the I^{2}C bus. Please contact XIA to discuss this development.

3.7 Interface to Host Computer/PDA

__NOTE:__ As of August 2009, the IDMA parallel communications interface is now implemented, in order to communicate with the new MicroComU companion board. The RS-232 communications port has always been supported. The SPORT interface is still not implemented at this time.

The microDXP interfaces to a computer or PDA via one of two connectors: the standard flex-cable port or the high-speed DMA port. Currently only the flex-cable port is supported.

3.7.1 Flex Cable Interface

A 0.5mm-pitch flex-cable provides the connection to power, serial communications and auxiliary digital lines. The flex cable provides for two dimensions of freedom, but does require alignment along the axis that bisects all of the contacts. Please refer to Appendix D for connector locations and pinouts.

Flex Cable Interface Resources		
Resource	*Function*	*Description*
RS-232	low-rate serial communications	Default communications interface.
I^2C	low-rate serial communications	Peripheral device interface, e.g. indicators, DACs, etc.
SPORT	mid-rate serial communications	Alternate serial communications interface, e.g. for multi-channel systems that require moderate readout bandwidth.
AUX0-3	Reserved	Auxiliary digital I/O lines. Connect to FiPPI.
GATE*	DAQ control	Inhibits data acquisition when low.
EXTINT*	DSP interrupt	Extra interrupt line, active low.

3.7.2 High Speed Interface

A high-density board-to-board connection is also included on the microDXP. The so-called high-speed interface includes all resources carried on the flex-cable interface, plus Direct Memory Access (DMA) to the DSP. It was included for applications requiring very fast data transfer rates.

High Speed (board-to-board) Interface Resources		
Resource	*Function*	*Description*
DMA	High-rate parallel communications	Direct Memory Access to DSP memory, for the highest bandwidth data transfers.
RS-232	Low-rate serial communications	Default communications interface.
I^2C	Low-rate serial communications	Peripheral device interface, e.g. indicators, DACs, etc.
SPORT	Mid-rate serial communications	Alternate serial communications interface, e.g. for multi-channel systems that require moderate readout bandwidth.

AUX0-3	Reserved	Auxiliary digital I/O lines. Connect to FiPPI.
GATE*	DAQ control	Inhibits data acquisition when low.
EXTINT*	DSP interrupt	Extra interrupt line, active low.

4 Digital Filtering: Theory of Operation and Implementation Methods

This chapter provides an in-depth discussion of x-ray pulse-processing theory both generally and as implemented in the microDXP. The topics include how digital filters work, x-ray detection, thresholds, baselines, pileup inspection, and input and output count rates. Topics are covered to illustrate the theoretical issues, practical implementation, and how to adjust parameters to obtain best performance.

The acronym DXP stands for “Digital X-ray Processor” and refers to XIA’s standard digital processing technology, which is included in many XIA products, including the microDXP.

4.1 X-ray Detection and Preamplifier Operation

Energy dispersive detectors, which include such solid state detectors as Si(Li), HPGe, HgI_2, CdTe and CZT detectors, are generally operated with charge sensitive preamplifiers. When an x-ray is absorbed in the detector material it releases an electric charge $Q_x = E_x/\varepsilon$, where the material constant ε is the amount of energy needed to form an electron-hole pair. Q_x is integrated onto the preamplifier’s feedback capacitor C_f, to produce the voltage $V_x = Q_x/C_f = E_x/(\varepsilon C_f)$. Measuring the energy E_x of the x-ray therefore requires a measurement of the voltage step V_x in the presence of the amplifier’s noise σ. Figure 4.1 and Figure 4.3 depict reset-type and RC-type charge sensitive amplifiers, respectively. In both figures the detector D is biased by voltage source HV (either positive or negative) and connected to the input of amplifier A. Note that the *signal polarity* must be distinguished from the *bias voltage polarity*. The signal polarity is positive if the voltage step V_x is a rising edge, as displayed in Figure 4.1. Whether signal polarity is positive or negative depends upon the preamplifier’s design and does not depend upon bias voltage polarity, which is specified on the detector and is determined by its design.

4.1.1 Reset-Type Preamplifiers

Figure 4.3a is a simplified schematic of a reset-type preamplifier, wherein C_f is discharged through the switch S from time to time when the amplifier’s output voltage gets so large that it behaves nonlinearly. Switch S may be an actual transistor switch, or may operate equivalently by another mechanism. In pulsed optical reset preamps light is directed at the amplifier A’s input FET to cause it to discharge C_f. In transistor reset preamps, the input FET may have an additional electrode which can be pulsed to discharge C_f. The output of a reset-type preamplifier following the absorption of an x-ray of energy E_x in detector D is a voltage step of amplitude V_x. Two x-ray steps are shown in Figure 4.3b as a step.

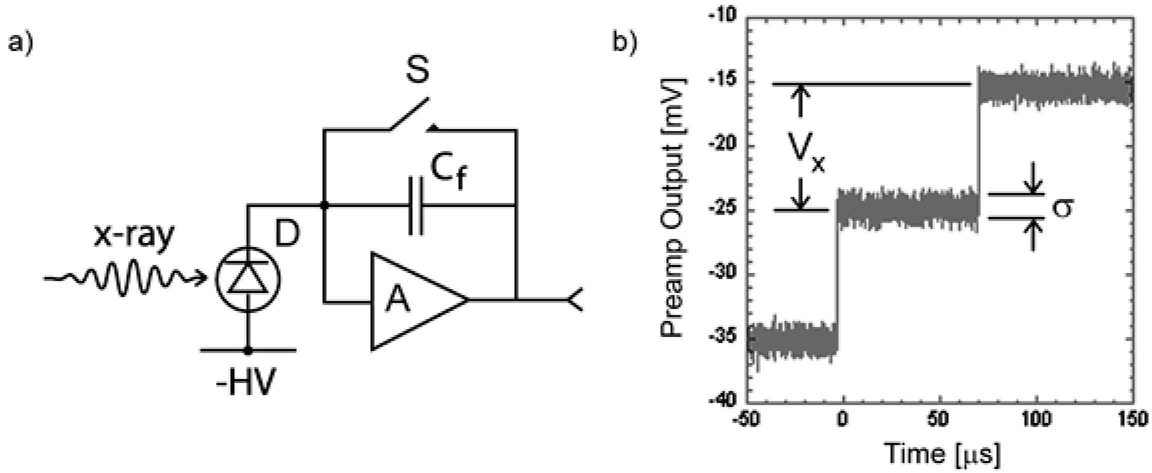

Figure 4.1: **a)** Reset-type charge sensitive preamplifier with a negatively biased detector; **b)** Output on absorption of x-ray rays. Note that the steps have a rising edge, so that the signal polarity is positive.

Figure 4.2 depicts the large-signal saw tooth waveform that results from successive x-ray steps followed by the reset. Note that the units here are Volts and milliseconds vs. millivolts and microseconds in the previous figure.

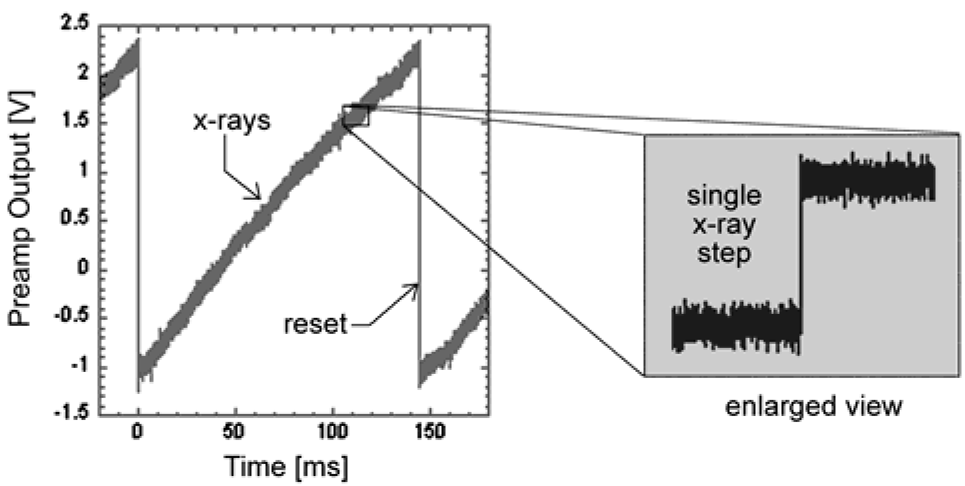

Figure 4.2: The large-signal reset waveform for a reset-type preamplifier with positive signal polarity, as displayed on a real oscilloscope. Note that the large signal character is not displayed in the microDXP diagnostic ADC readout, e.g. used in microManager's "ADC Trace" diagnostic tool, looks quite different because of the dynamic range reduction carried out in the ASC, as described in §3.2.1.

4.1.2 RC-Type Preamplifiers

Figure 4.3a is a simplified schematic of an RC-type preamplifier, wherein C_f is discharged continuously through feedback resistor R_f. The output of an RC-type preamplifier following the absorption of an x-ray of energy E_x in detector D is, again, a voltage step of amplitude V_x. The continuous discharge of C_f through R_f results in an exponential voltage decay after the x-ray step, with decay constant τ, where:

$$\tau = R_f C_f$$

Equation 4-1

In practice the decay time may depend on subsequent circuitry, i.e. if a pole-zero cancellation circuit is used, thus τ may not be directly related to the feedback elements of the front-end. The point of this simplified model is that the resulting waveform is a single-pole RC decay. The discussion in § 4.2 through § 0 assumes a reset-type preamplifier, but is mostly applicable to RC-type preamplifiers. § 4.7 describes the few key differences in the processing of RC-type preamplifier signals.

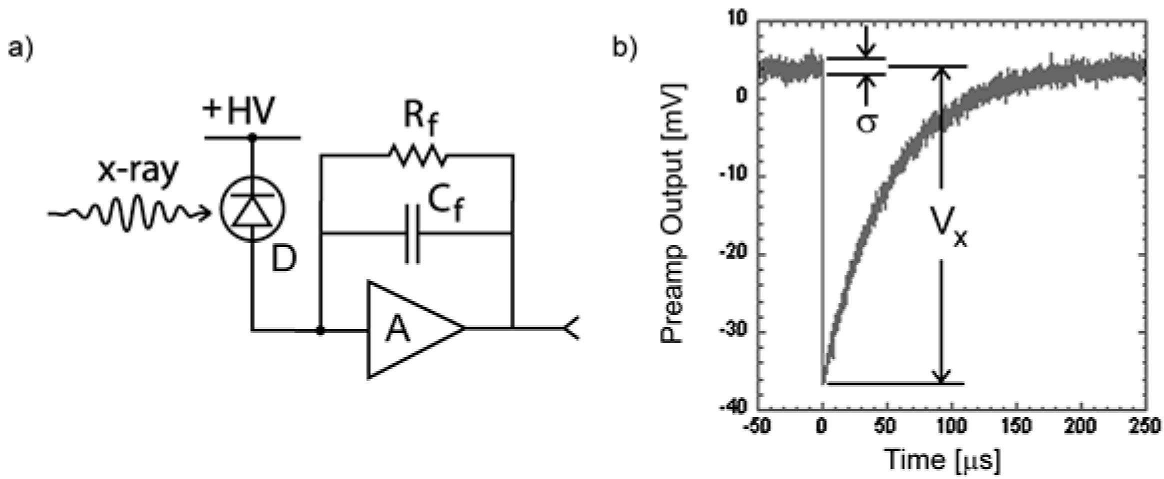

Figure 4.3: **a)** RC-type charge sensitive preamplifier with a positively biased detector; **b)** Output on absorption of an x-ray. Note that the step has a falling edge, thus the signal polarity is negative.

4.2 X-ray Energy Measurement & Noise Filtering

Reducing noise in an electrical measurement is accomplished by filtering. Traditional analog filters use combinations of a differentiation stage and multiple integration stages to convert the preamp output steps, such as shown in Figure 4.1b, into either triangular or semi-Gaussian pulses whose amplitudes (with respect to their baselines) are then proportional to V_x and thus to the x-ray's energy.

Digital filtering proceeds from a slightly different perspective. Here the signal has been digitized and is no longer continuous, but is instead a string of discrete values, such as shown in Figure 4.4. The data displayed are actually just a subset of Figure 4.3b, which was digitized by a Tektronix 544 TDS digital oscilloscope at 10 MSA (million samples per second). Given this data set, and some kind of arithmetic processor, the obvious approach to determining V_x is to take some sort of average over the points before the step and subtract it from the value of the average over the points after the step. That is, as shown in Figure 4.4, averages are computed over the two regions marked "Length" (the "Gap" region is omitted because the signal is changing rapidly here), and their difference taken as a measure of V_x. Thus the value V_x may be found from the equation:

$$V_{x,k} = -\sum_{i\,(\text{before})} w_i v_i + \sum_{i\,(\text{after})} w_i v_i$$

Equation 4-2

Where the values of the weighting constants w_i determine the type of average being computed. The sums of the values of the two sets of weights must be individually normalized.

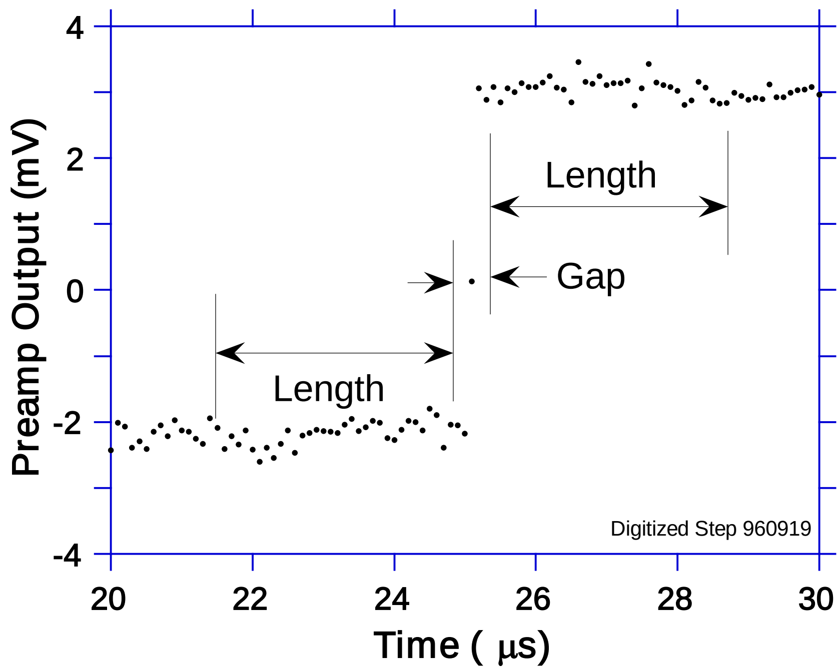

Figure 4.4: Digitized version of one of the x-ray steps of Figure 4.3b.

The primary differences between different digital signal processors lie in two areas: what set of weights $\{w_i\}$ is used and how the regions are selected for the computation of Equation 4-2. Thus, for example, when the weighting values decrease with separation from the step, then the equation produces “cusp-like” filters. When the weighting values are constant, one obtains triangular (if the gap is zero) or trapezoidal filters. The concept behind cusp-like filters is that, since the points nearest the step carry more information about its height, they should be more strongly weighted in the averaging process. How one chooses the filter lengths results in time variant (the lengths vary from pulse to pulse) or time invariant (the lengths are the same for all pulses) filters. Traditional analog filters are time invariant. The concept behind time variant filters is that, since the x-rays arrive randomly and the lengths between them vary accordingly, one can make maximum use of the available information by adjusting Length on a pulse-by-pulse basis.

In principal, the very best filtering is accomplished by using cusp-like weights and time variant filter length selection. There are serious costs associated with this approach however, both in terms of computational power required to evaluate the sums in real time and in the complexity of the electronics required to generate (usually from stored coefficients) normalized $\{w_i\}$ sets on a pulse-by-pulse basis. A few such systems have been produced but typically cost about $13K per channel and are count rate limited to about 30

Kcps. Even time invariant systems with cusp-like filters are still expensive due to the computational power required to rapidly execute strings of multiply and adds. One commercial system exists which can process over 100 Kcps, but it too costs over $12K per channel.

The DXP processing system developed by XIA takes a different approach because it was optimized for very high-speed operation and low cost per channel. It implements a fixed length filter with all w_i values equal to unity and in fact computes this sum afresh for each new signal value k. Thus the equation implemented is:

$$L\,V_{x,k} = -\sum_{i=k-2L-G+1}^{k-L-G} v_i + \sum_{i=k-L+1}^{k} v_i$$

Equation 4-3

where the filter length is L and the gap is G. The factor L multiplying Vx,k arises because the sum of the weights here is not normalized. Accommodating this factor is trivial for the DXP's host software. The operations are carried out using hardwired logic in a field programmable gate array (FPGA) that is called the FiPPI because is implements Filtering, Peak capture, and Pileup Inspection.

In the FiPPI, Equation 4-3 is actually implemented by noting the recursion relationship between Vx,k and Vx,k-1, which is:

$$L\,V_{x,k} = L\,V_{x,k-1} + v_k - v_{k-L} - v_{k-L-G} + v_{k-2L-G}$$

Equation 4-4

***Decimation** by N means to break up the data into sequential sums of length $D = 2^N$.*

Peaking Time** ranges vs **Decimation N** **(8MHz clock assumed)	
N = 0	*0.75 – 3.00 μs*
N = 1	*1.50 – 6.00 μs*
N = 2	*3.00 – 12.0 μs*
N = 3	*6.00 – 24.0 μs*
N = 4	*12.0 – 48.0 μs*

While this relationship is very simple, it is still very effective. In the first place, this is the digital equivalent of triangular (or trapezoidal if G = 0) filtering which is the analog industry's standard for high rate processing. In the second place, one can show theoretically that if the noise in the signal is white (i.e. Gaussian distributed) above and below the step, which is typically the case for the short shaping times used for high signal rate processing, then the average in Equation 4-4 actually gives the best estimate of V_x in the least squares sense. This, of course, is why triangular filtering has been preferred at high rates. Triangular filtering with time variant filter lengths can, in principle, achieve both somewhat superior resolution and higher throughputs but comes at the cost of a significantly more complex circuit and a rate dependent resolution, which is unacceptable for many types of precise analysis. In practice, XIA's design has been found to duplicate the energy resolution of the best analog shapers while approximately doubling their throughput, providing experimental confirmation of the validity of the approach.

A practical limitation on the implementation of Equation 4-4 is that two FIFO memories are required, one of length L and one of Length L+G. Since memory space is limited in FPGAs, we have restricted our designs to values of L+G less than 32. Since the microDXP samples at 20 MSA, this corresponds to a peaking time of about 1.5 μs, a significant limitation. XIA overcomes this limitation by first "decimating" the data stream from the ADC by performing sequential sums of D data points, where $D = 2^N$. We refer to this filtering procedure as "Decimating by N". By feeding the decimated data in an Equation 4-4 filter, we now obtain peaking times that are extended to L*D. It is important

to understand that no data are lost in this procedure, we have merely rearranged the order of the summations represented in Equation 4-3. By extension, a "Decimation N FiPPI" is one that decimates the data by N before applying the energy filter. The common decimation values in the microDXP are 0, 2, and 4, corresponding to averaging times of 125 ns, 500 ns, and 2.0 μs. The associated peaking time ranges are 0.75 – 3.0 μs, 3.0 – 12.0 μs, and 12.0 – 48.0 μs. *Note:* This assumes the standard 8MHz pipeline clock. For a 16MHz pipeline clock (optional), the averaging times and peaking time ranges are exactly one-half of above values.

4.3 Trapezoidal Filtering in the DXP

From this point onward, we will only consider trapezoidal filtering as it is implemented in the DXP according to Equation 4-3 and Equation 4-4. The result of applying such a filter with Length L = 20 and Gap G = 4 to the same data set of Figure 4.4 is shown in Figure 4.5. The filter output V_X is clearly trapezoidal in shape and has a risetime equal to L, a flattop equal to G, and a symmetrical falltime equal to L. The basewidth, which is a first-order measure of the filter's noise reduction properties, is thus 2L+G. This raises several important points in comparing the noise performance of the DXP to analog filtering amplifiers. First, semi-Gaussian filters are usually specified by a *shaping time*. Their peaking time is typically twice this and their pulses are not symmetric so that the basewidth is about 5.6 times the shaping time or 2.8 times their peaking time. Thus a semi-Gaussian filter typically has a slightly better energy resolution than a triangular filter of the same peaking time because it has a longer filtering time. This is typically accommodated in amplifiers offering <u>both</u> triangular and semi-Gaussian filtering by stretching the triangular peaking time a bit, so that the *true* triangular peaking time is typically 1.2 times the selected semi-Gaussian peaking time. This also leads to an apparent advantage for the analog system when its energy resolution is compared to a digital system with the same nominal peaking time.

One extremely important characteristic of a digitally shaped trapezoidal pulse is its extremely sharp termination on completion of the basewidth 2L+G. This may be compared to analog filtered pulses which have tails which may persist up to 40% of the peaking time, a phenomenon due to the finite bandwidth of the analog filter. As we shall see below, this sharp termination gives the digital filter a definite rate advantage in pileup free throughput.

In practice it is also important to realize that implementing an energy filter in a Decimation N FiPPI sets certain limitations on the practical flat-top lengths that can be obtained in trapezoidal filters. Because the decimation process is uncorrelated with the arrival of x-rays, the gap G must be 3 or greater to assure that the filter's peak truly represents the x-ray's energy. Therefore, the minimum Decimation N gap time is $G*2^N*\Delta t$, where Δt is the ADC's sampling interval. With the microDXP's Δt = 125 ns sampling interval, for instance, the smallest useful flat-top in Decimation 4 is 6.0 μs. *Note:* Again, this assumes the standard 8MHz pipeline clock. For a 16MHz pipeline clock (optional), the sampling interval and gap time would be exactly one-half of above values.

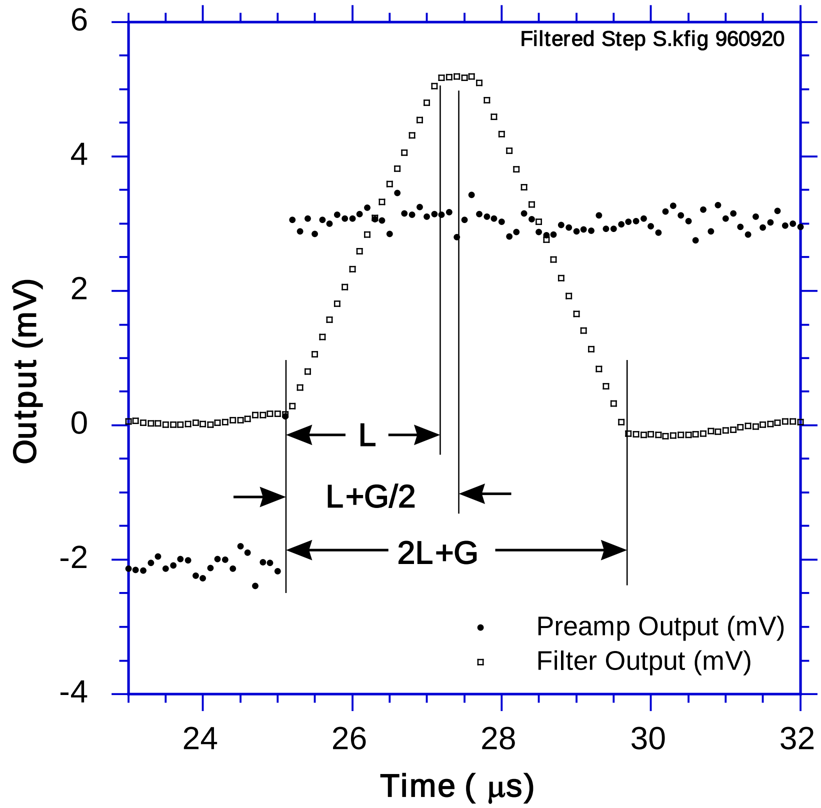

Figure 4.5: Trapezoidal filtering the Preamp Output data of Figure 4.4 with L = 20 and G = 4.

4.4 Baseline Issues

4.4.1 The Need for Baseline Averaging

Figure 4.6 shows the same event as is Figure 4.5 but over a longer time interval to show how the filter treats the preamplifier noise in regions when no x-ray pulses are present. As may be seen, the effect of the filter is both to reduce the amplitude of the fluctuations and reduce their high frequency content. This signal is termed the *baseline* because it establishes the reference level or offset from which the x-ray peak amplitude V_X is to be measured. The fluctuations in the baseline have a standard deviation σ_e which is referred to as the *electronic noise* of the system, a number which depends on the peaking time of the filter used. Riding on top of this noise, the x-ray peaks contribute an additional noise term, the *Fano noise*, which arises from statistical fluctuations in the amount of charge Q_X produced when the x-ray is absorbed in the detector. This Fano noise σ_f adds in quadrature with the electronic noise, so that the total noise σ_t in measuring V_X is found from

$$\sigma_t = \mathrm{sqrt}(\ \sigma_f^2 + \sigma_e^2)$$

Equation 4-5

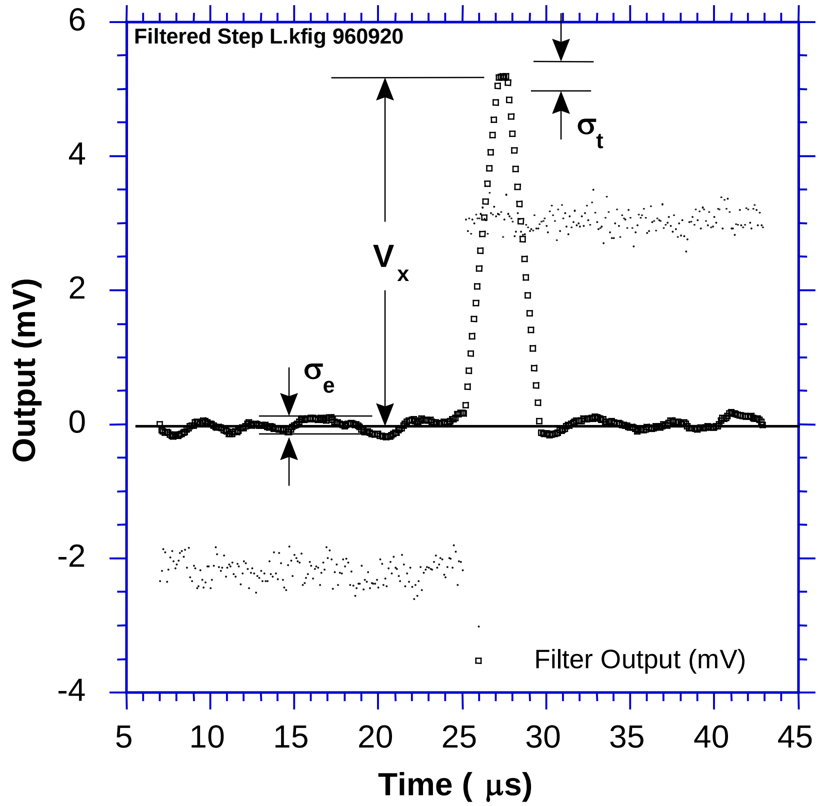

Figure 4.6: The event of Figure 4.5 displayed over a longer time period to show baseline noise.

The Fano noise is only a property of the detector material. The electronic noise, on the other hand, may have contributions from both the preamplifier and the amplifier. When the preamplifier and amplifier are both well designed and well matched, however, the amplifier's noise contribution should be essentially negligible. Achieving this in the mixed analog-digital environment of a digital pulse processor is a non-trivial task, however.

In the general case, the mean baseline value is not zero. This situation arises whenever the slope of the preamplifier signal is not zero between x-ray pulses. This can be seen from Equation 4-3. When the slope is not zero, the mean values of the two sums will differ because they are taken over regions separated in time by L+G, on average. Such non-zero slopes can arise from various causes, of which the most common is detector leakage current.

When the mean baseline value is not zero, it must be determined and subtracted from measured peak values in order to determine V_X values accurately. If the error introduced by this subtraction is not to significantly increase σ_t, then the error in the baseline estimate σ_b must be small compared to σ_e. Because the error in a single baseline measurement is σ_e, by definition, this means that multiple baseline measurements will have to be averaged. This number, N_B is the Baseline Average. For example, if N_B = 128 measurements are averagedthen the total noise will be as shown in Equation 4-6.

$$\sigma_t = \mathrm{sqrt}(\ \sigma_f^2 + (1+1/128)\sigma_e^2)$$

Equation 4-6

This results in less than 0.5 eV degradation in resolution, even for very long peaking times, when resolutions of order 130 eV are obtained.

4.4.2 Raw Baseline Measurement

*Applying a **Baseline Cut** can improve performance when the Baseline Histogram is non-Gaussian. Outlying data points are'cut' from the running **Baseline Average** (though still included in the histogram)*

The output of the *baseline filter* (either the energy filter itself, or a derivative of it) is sampled periodically in the explicit absence of an x-ray step, defined by a baseline threshold. In practice, the DXP initially makes a series of N_B baseline measurements to compute a starting baseline mean. It then makes additional baseline measurements at quasi-periodic intervals to keep the estimate up to date. These values are stored internally and can be read out to construct a spectrum of baseline noise, referred to as the Baseline Histogram. This is recommended because of its excellent diagnostic properties. When all components in the spectrometer system are working properly, the baseline spectrum should be Gaussian in shape with a standard deviation reflecting σ_n. Deviations from this shape indicate various pathological conditions which may also cause the x-ray spectrum to be distorted and therefore have to be fixed.

The situation is remedied by removing ("cutting") outlying samples from the baseline average described below. If the maximum in the baseline distribution lies at E_0, then captured baseline values that deviate from E_0 by more than ΔE^+ and ΔE^-, respectively, are not included in the running baseline average. Note that *all captured baseline values are included in the Baseline Histogram*, however, so that it is always a valid representation of the system's behavior.

4.4.3 Baseline Averaging in the DXP

A running average of baseline measurements is computed, which is then subtracted from sampled peak values to compute the energy of corresponding incident x-rays. The number of baseline samples averaged is set in microManager as "Number of Baselines to Average". In the DSP this is converted into the parameter BLFILTER according to the equation:

baseline samples averaged = 32768 / BLFILTER

Decimation	**# Baseline Samples to Average**	**BLFILTER** (DSP Parameter)
0	64	512
2	128	256
4	256	128

Table 4.1: Typical values used for baseline averaging. The best value for each decimation should be determined empirically, though the general trend illustrated in the table, i.e. larger number to average for higher decimations, should be followed.

Physically, the baseline is a measure of the instantaneous slope (volts/sec) for a pulsed-reset detector, and a measure of the DC offset for an RC-feedback preamplifier. For a perfect detector and preamplifier the baseline value is independent of time. In fact, the variation in leakage current of the

detector and offset drift and 1/f noise of the preamplifier contribute to a baseline value that wanders at low frequencies. The goal is to achieve a baseline average that has a sufficient number of samples to average out the high frequency noise, but which still reflects the 'local' instantaneous baseline. Generally speaking, Number of Baselines to Average is set to achieve the best energy resolution performance over the desired range of input count rate. There are two considerations worth emphasizing:

1. *Excess detector/preamplifier noise and pickup (all decimations):* The values in the table above implicitly assume a flat noise spectrum from the preamplifier. A high-frequency noise peak can result in poor relative performance at the corresponding 'resonant' peaking time. Often this problem can be mediated, though not eliminated, by *increasing* the number of baseline samples in the average for the affected peaking times. On the other hand, excess low-frequency noise, i.e. wandering, can be remedied by *reducing* the number baseline samples in the average.
2. *High rate performance (decimation 0):* At higher rates, i.e. > 50% deadtime, the slow filter returns less and less often to baseline, thus the time between baseline samples grows longer. This is the primary cause of degraded energy resolution at high rates. Firmware of decimation 2 and above now employs a proprietary circuit that virtually eliminates this problem, resulting in industry-leading count rate stability. This improvement cannot however be implemented in the decimation 0 firmware. The resolution can nonetheless be improved in most cases by *reducing* the number of baseline samples in the average.

4.5 X-ray Detection & Threshold Setting

Before capturing a value of V_X we must first detect the x-ray. X-ray steps (in the preamp output) are detected by digitally comparing the output of a trapezoidal filter to a threshold.

In the DXP up to three trapezoidal filters are implemented: *fast*, *intermediate* and *slow;* each with a threshold that can be individually enabled or disabled. A fast filter very quickly detects larger x-ray steps. A slow (energy) filter averages out the most noise and can thus detect smaller x-ray steps, but has a response that is much slower. An intermediate filter (used in decimations 2 and 4 only) is a derivative of the slow filter that provides a balance between the speed of the fast filter and the noise reduction of the slow filter.

The fast filter is used solely for x-ray detection, i.e. a threshold crossing initiates event processing. Its short basewidth (2L+G) means that successive pulses that would 'pile-up' in slower filters can be resolved in the fast filter and rejected from the spectrum (see Figure 4.11 below). Conversely, little noise reduction is achieved in the fast filter, thus the fast threshold cannot be set to detect particularly low x-ray energies.

The intermediate filter is used for all decimations other than 0. Its threshold is applied as part of the baseline acquisition circuitry, i.e. baseline measurements are taken when the signal is *below* this threshold. Intermediate threshold crossings by default also trigger event processing, extending the detectable energy range significantly below the fast filter threshold. Note that this threshold is initialized to the maximum, i.e. most conservative, value, and should be adjusted downward by the user for best performance.

After an x-ray has been detected, the step height is measured at the slow filter output. Although its excellent noise reduction also allows *detection* of the very lowest energy x-rays, its slow response precludes an accurate determination of pulse pileup. For this reason the slow threshold should be disabled in almost all cases.

4.6 Peak Capture Methods

As noted above, we wish to capture a value of V_X for each x-ray detected and use these values to construct a spectrum. This process is also significantly different between digital and analog systems. In the analog system the peak value must be "captured" into an analog storage device, usually a capacitor, and "held" until it is digitized. Then the digital value is used to update a memory location to build the desired spectrum. During this analog to digital conversion process the system is dead to other events, which can severely reduce system throughput. Even single channel analyzer systems introduce significant deadtime at this stage since they must wait some period (typically a few microseconds) to determine whether or not the window condition is satisfied.

Digital systems are much more efficient in this regard, since the values output by the filter are already digital values. All that is required is to capture the peak value – it is immediately ready to be added to the spectrum. If the addition process can be done in less than one peaking time, which is usually trivial digitally, then no system deadtime is produced by the capture and store operation. This is a significant source of the enhanced throughput found in digital systems.

Once an active threshold is exceeded, the microDXP employs one of two methods to capture the slow energy filter output such that the best measure of V_x results. For decimations other than 0 the slow filter output is monitored over a finite interval of time in the region of its maximum, and the *maximum value within that interval* is captured. This method is referred to as "peak finding" or "max capture". For decimation 0, the slow filter is *sampled at a fixed time interval* after the pulse is detected by the fast filter. This method is referred to as "peak sampling".

After describing in §4.6.1 below how to set the Gap parameter so that there will be a quality value of the energy filter to capture, we describe the two methods in detail in §4.6.2.

4.6.1 The Slow Filter Gap Length

When starting with a new detector, it is important first to set SLOWGAP to a minimum of 3, and *at least one unit greater than* the smallest value, in decimated clock cycles (see Table 4.2), that encloses the entire preamplifier risetime, per §5.10.2.

Decimation	# ADC Samples averaged	Decimated Clock frequency	Decimated Clock cycle interval	Peaking Time Range (in μs)
0	1	8 MHz	125 ns	0.75 – 3
1	2	4 MHz	250	1.5 – 6
2	4	2 MHz	500 ns	3 - 12
3	8	1 MHz	1.0 μs	6 - 24
4	16	500 kHz	2.0 μs	12 - 48

Table 4.2: For decimation 0 the slow filter output is sampled a fixed time after the x-ray is detected. PEAKSAM must be set properly to achieve optimum performance.

For example, consider a preamplifier with a pulse risetime of 460ns. For decimations 1, 2, 3, and 4 SLOWGAP would be set to 3 or greater. For decimation 0 SLOWGAP would be set to 5 or greater.

4.6.2 Peak Sampling vs. Peak Finding

The figures below demonstrate the two approaches. For decimations 2 and 4 the slow filter output is monitored over a finite interval of time, and the *maximum value within that interval* is selected. This method is referred to as 'peak finding'. The interval is set automatically, solely based on the values of the DXP parameters SLOWLEN and PEAKINT. SLOWLEN and PEAKINT are both automatically derived from the peaking time value selected in microManager and should normally not be adjusted by the user. PEAKINT is also a pileup inspection parameter, as will be discussed in further detail in §4.8.

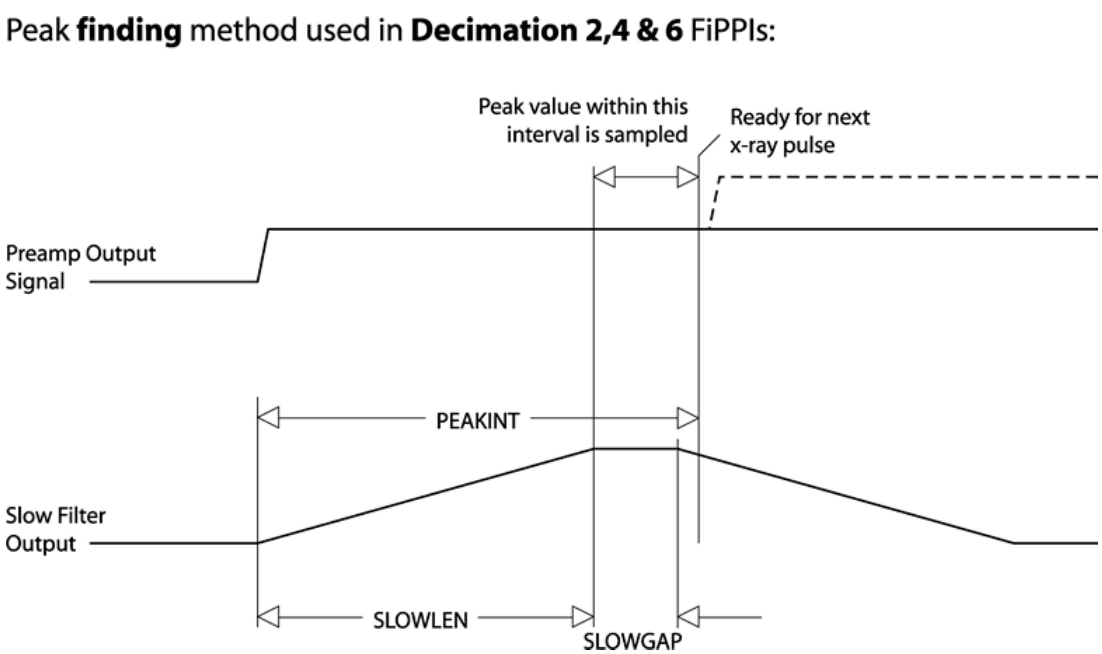

Figure 4.7 For decimations 2 and 4 the slow filter output is monitored and the peak value is selected.

For decimation 0, the slow filter output is instead sampled a *fixed time* after the x-ray is detected. This method is referred to as 'peak sampling'. An

additional 'Peak Sampling' timer is started when an x-ray step is detected which expires after PEAKSAM decimated clock cycles. PEAKSAM must be less than PEAKINT, and should typically be set such that the sample point lies in the 'flat-top' region of the slow filter output:

$$\text{SLOWLEN} \leq \text{PEAKSAM} \leq \text{SLOWLEN} + \text{SLOWGAP}$$

Equation 4-7

The precise setting has a strong effect on energy resolution and should be determined empirically for each new detector. More on this below...

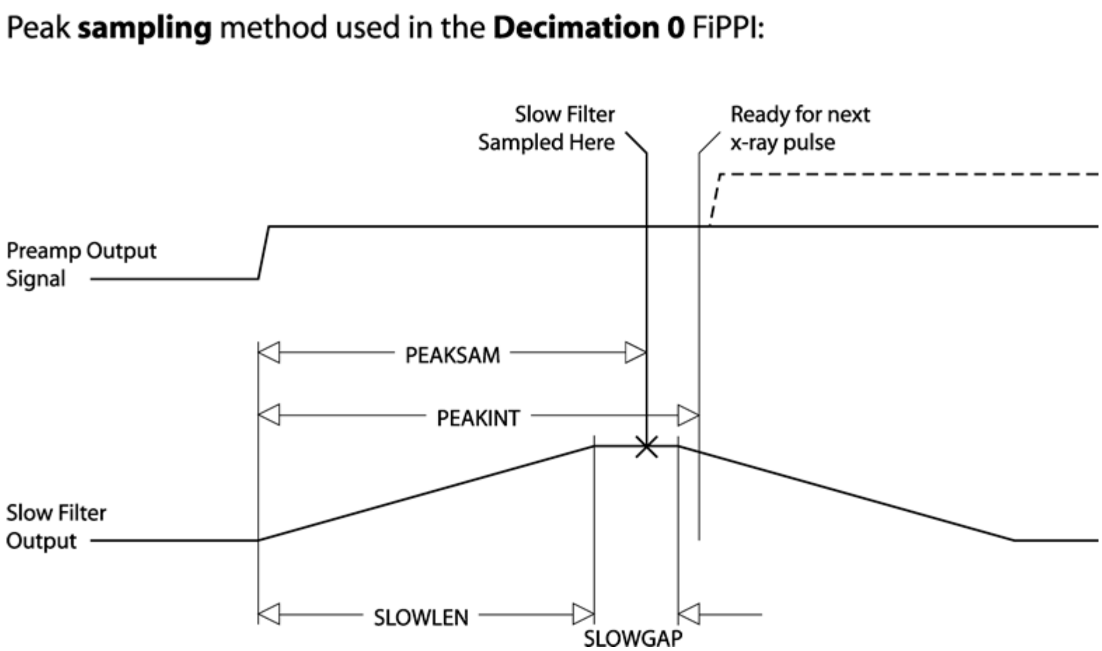

Figure 4.8: For decimation 0 the slow filter output is sampled a fixed time after the x-ray is detected. PEAKSAM must be set properly to achieve optimum performance.

In our experience values at the low end (i.e. PEAKSAM ~ SLOWLEN) tend to work better. We recommend that you record the initial value of PEAKSAM and then change it in steps of 1, working out from the initial value. Certain PEAKSAM values may cause the Saturn to crash. Do not be alarmed, just restart and be sure to enter a valid PEAKSAM value before proceeding. Making a plot of energy resolution versus PEAKSAM will indicate the best value to select.

This determination need only be done for one peaking time per decimation. The result can then be applied to any value of SLOWLEN and SLOWGAP using the following recipe:

$$\text{PEAKSAM} = (\text{SLOWLEN} + \text{SLOWGAP}) - X$$

Equation 4-8

4.7 Energy Measurement with Resistive Feedback Preamplifiers

In previous sections, the pulse height measurement was shown for the case of reset-type preamplifiers. The reset-type scheme is most often used for optimum energy resolution x-ray detectors. Other detectors use an RC-type preamplifier, as described in § 4.1.2. Resistive feedback is most often used for gamma-ray detectors that cover a larger dynamic range and where the electronic noise is not as significant a contribution to energy resolution.

Where analog shaping amplifiers typically have a “pole-zero” adjustment to cancel out the exponential decay, the DXP uses a patented exponential decay correction to achieve good energy resolution without a pole-zero correction. Figure 4.9 and Figure 4.10 illustrate the method used. The first shows the output voltage of a RC feedback preamplifier with a x-ray or γ-ray step of amplitude A appearing at t=0. V_e is the voltage just before the step pulse arrives and V_0 is the asymptotic value that the signal would decay to in the absence of steps. t_1 is the earliest time used in the slow filter, L and G are the length and gap of the trapezoidal filter in clock units, and Δt is the clock period, In addition to the normal slow filter measurement of the step height, the ADC amplitude, V_D is made at time t_D. In the following discussion, it is assumed that the signal rise-time is negligible.

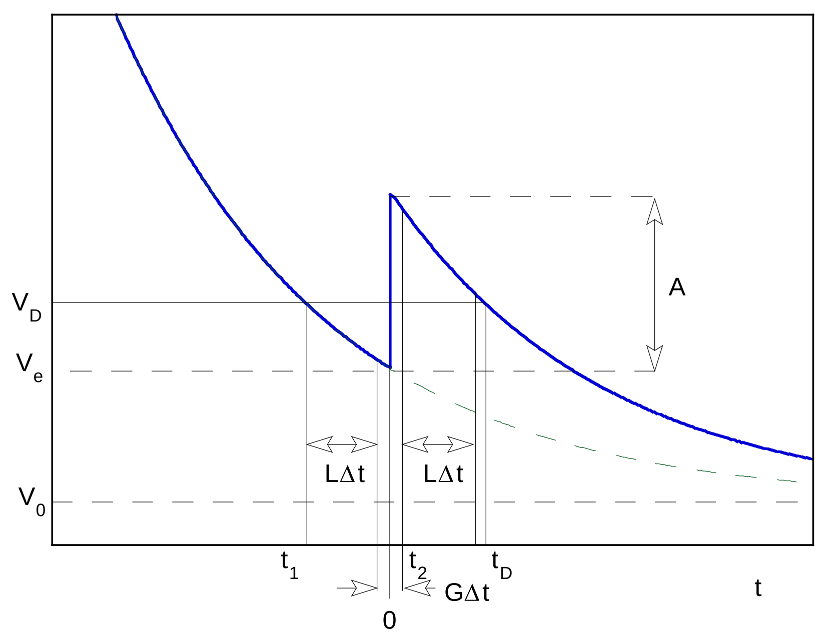

Figure 4.9: RC preamplifier output voltage. An x-ray step of amplitude A occurs at time t=0.

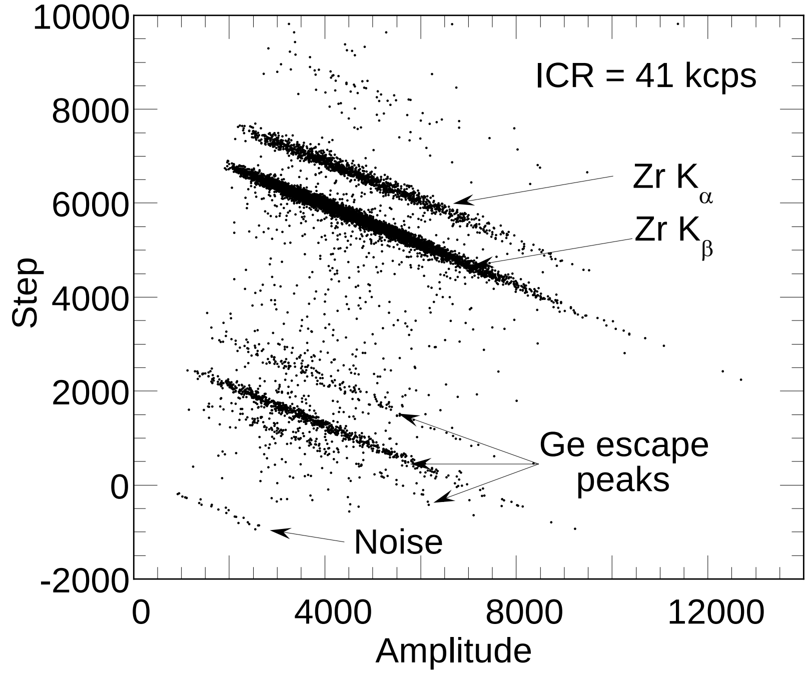

Figure 4.10: Correlation between step size and amplitude for Zr Kα x-ray events measured with the DXP-4C.

As Figure 4.10 makes clear, there is a linear correlation between the step height from the trapezoidal filter and the ADC amplitude, for pulses of a given energy. This is due to the fact that the exponential decay causes a deficit in the measured step height, which grows linearly with the distance from the asymptotic ADC offset at zero count rate.

The DSP reads these two values for each event that passes the FiPPI's trigger criteria, and makes a correction of the form:

$$E = k_1 (S_X + k_2 V_X - < S_B + k_2 V_B >)$$

Equation 4-9

Here the quantities S_X and V_X are the step height and ADC amplitude measured for the step, and the corresponding values with the B subscript are "baseline" values, which are measured frequently at times when there is no trigger. The brackets <> indicate that the baseline values are averaged over a large enough number of events to not introduce additional noise in the measurement. The constant k_2 (the DSP parameter called RCFCOR) is inversely proportional to the exponential decay time; this correction factor is a constant for a detector channel at a fixed gain and shaping time. The constant k_1 is effectively a gain factor, and is taken into account with a detector gain calibration.

The parameter RCFCOR is a function of the digital filter parameters (SLOWLEN, SLOWGAP and DECIMATION) and the preamplifier decay time (the DSP parameter TAURC). The user-entered decay time TAURC is in units of 50 ns clock ticks. At the start of an acquisition run, the DSP calculates RCFCOR using the following approximate expression:

$$RCFCOR = 2^{DEC} * (LEN + GAP) / (TAURC – (LEN + GAP/2 + 3)*2^{DEC})$$

Equation 4-10

The above expression is valid for peaking times less than about TAURC/2. Alternatively, RCFCOR can be determined empirically in a special test run from a linear fit of data, as in Figure 4.10.

4.8 Pile-up Inspection

The captured value V_X (see Figure 4.6) will only be a valid measure of t\its associated x-ray's energy provided that its filtered pulse is sufficiently well separated in time from its preceding and succeeding neighbor pulses so that its peak amplitude is not distorted by the action of the trapezoidal filter on those neighbor pulses. That is, if the pulse is not *piled up*. The relevant issues may be understood by reference to Figure 4.11, which shows 5 x-rays arriving separated by various intervals.

Because the triangular filter is a linear filter, its output for a series of pulses is the linear sum of its outputs for the individual members in the series. In Figure 4.11 the pulses are separated by intervals of 3.2, 1.8, 5.7, and 0.7 μs, respectively. The fast filter has a peaking time of 0.4 μs with no gap. The slow filter has a peaking time of 2.0 μs with a gap of 0.4 μs.

The first kind of pileup is *slow pileup*, which refers to pileup in the slow channel. This occurs when the rising (or falling) edge of one pulse lies under the peak (specifically the sampling point) of its neighbor. Thus peaks 1 and 2 are sufficiently well separated so that the leading edge (point 2a) of peak 2 falls after the peak of pulse 1. Because the trapezoidal filter function is symmetrical, this also means that pulse 1's trailing edge (point 1c) also does not fall under the peak of pulse 2. For this to be true, the two pulses must be separated by at least an interval of L + G/2. Peaks 2 and 3, which are separated by only 1.8 μs, are thus seen to pileup in the present example with a 2.0 μs peaking time.

This leads to an important first point: whether pulses suffer slow pileup depends critically on the peaking time of the filter being used. The amount of pileup which occurs at a given average signal rate will increase with longer peaking times. We will quantify this in §4.10, where we discuss throughput.

Because the fast filter peaking time is only 0.4 μs, these x-ray pulses do not pileup in the fast filter channel. The DXP can therefore test for slow channel pileup by measuring for the interval PEAKINT after a pulse arrival time. If no second pulse occurs in this interval, then there is no trailing edge pileup. PEAKINT is usually set to a value close to L + G/2 + 1. Pulse 1 passes this test, as shown in the figure. Pulse 2, however, fails the PEAKINT test because pulse 3 follows in 1.8 μs, which is less than PEAKINT = 2.3 μs. Notice, by the symmetry of the trapezoidal filter, if pulse 2 is rejected because of pulse 3, then pulse 3 is similarly rejected because of pulse 2.

Pulses 4 and 5 are so close together that the output of the fast filter does not fall below the threshold between them and so they are detected by the pulse detector as only being a single x-ray pulse. Indeed, only a single (though somewhat distorted) pulse emerges from the slow filter, but its peak amplitude corresponds to the energy of neither x-ray 4 nor x-ray 5. In order to reject as

many of these fast channel pileup cases as possible, the DXP implements a fast channel pileup inspection test as well.

The fast channel pileup test is based on the observation that, to the extent that the risetime of the preamplifier pulses is independent of the x-rays' energies (which is generally the case in x-ray work except for some room temperature, compound semiconductor detectors) the basewidth of the fast digital filter (i.e. $2L_f + G_f$) will also be energy independent and will never exceed some maximum width MAXWIDTH. Thus, if the width of the fast filter output pulses is measured at threshold and found to exceed MAXWIDTH, then fast channel pileup must have occurred. This is shown graphically in the figure where pulse 3 passes the MAXWIDTH test, while the piled up pair of pulses 4 and 5 fail the MAXWIDTH test. Thus, in Figure 4.11, only pulse 1 passes both pileup inspection tests and, indeed, it is the only pulse to have a well defined flattop region at time PEAKSAMP in the slow filter output.

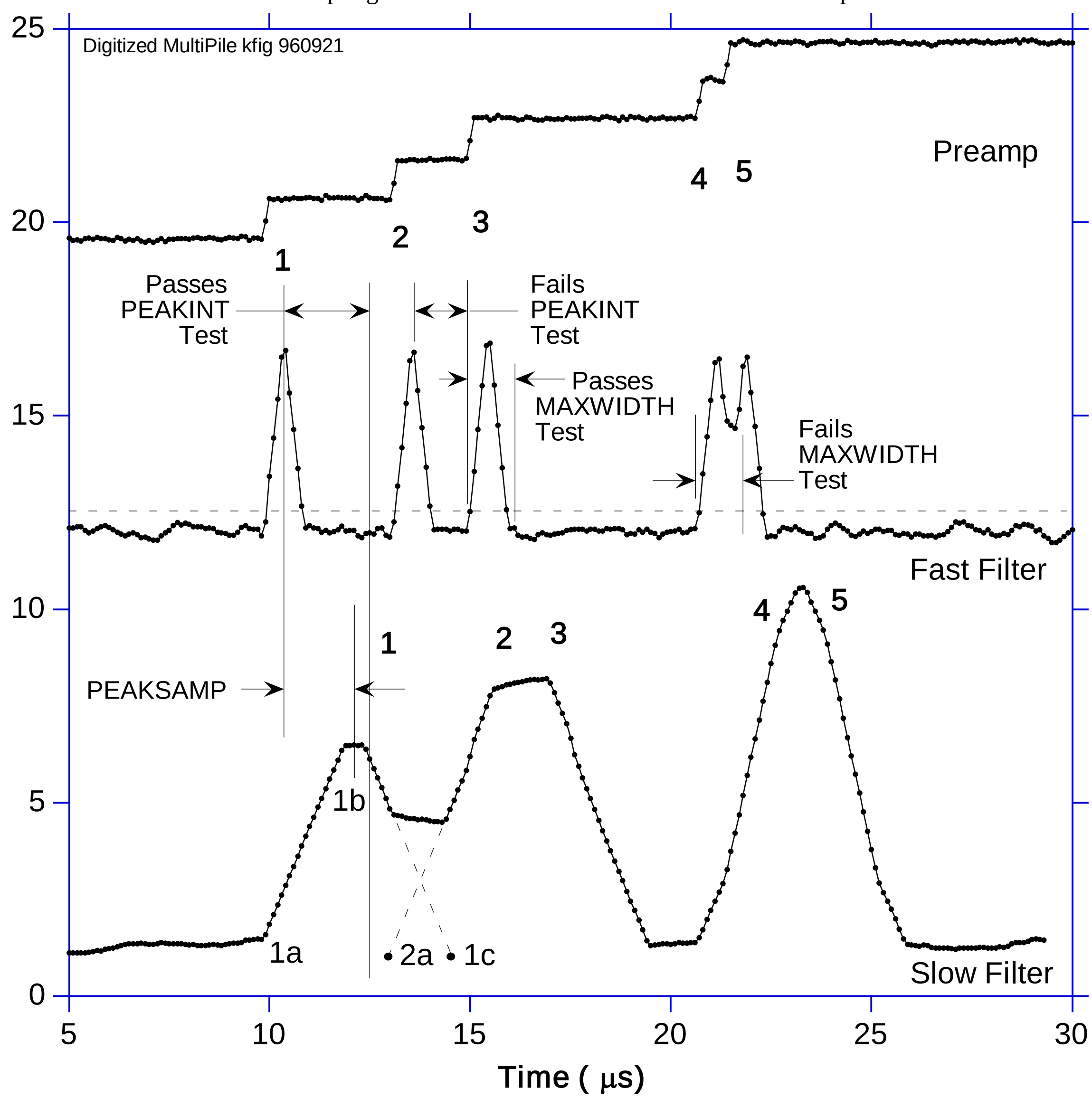

Figure 4.11: A sequence of 5 x-ray pulses separated by various intervals to show the origin of both slow channel and fast channel pileup and demonstrate how the two cases are detected by the DXP.

Note that PEAKINT and MAXWIDTH are both DSP parameters and are normally set automatically. In particular, there is almost never any benefit to a longer value of PEAKINT than the standard value as it does not improve energy resolution and only decreases throughput for a given input rate. Please see §5.10.5 for details on how to adjust MAXWIDTH.

4.9 Input Count Rate (ICR) and Output Count Rate (OCR)

During data acquisition, x-rays will be absorbed in the detector at some rate. This is the *true input count rate* , which we will refer to as ICR_t. Because of fast channel pileup, not all of these will be detected by the DXP's x-ray pulse detection circuitry, which will thus report a *measured input count rate* ICR_m, which will be less than ICR_t. This phenomenon, it should be noted, is a characteristic of all x-ray detection circuits, whether analog or digital, and is not specific to the DXP.

Of the detected x-rays, some fraction will also satisfy both fast and slow channel pileup tests and have their values of V_x captured and placed into the spectrum. This number is the *output count rate* , which we refer to as the OCR. The DXP normally returns, in addition to the collected spectrum, the REALTIME for which data was collected, the fast channel LIVETIME for which the fast channel was below threshold (and thus ready to detect a subsequent x-ray) together with the number FASTPEAKS of fast peaks detected and the number of V_x captured events EVTSINRUN. From these values, both the OCR and ICR_m can be computed according to Equation 4-11. These values can then be used to make deadtime corrections as discussed in the next section.

$$ICR_m = \text{FASTPEAKS/LIVETIME}; \quad \text{OCR} = \text{EVTSINRUN/REALTIME}$$

Equation 4-11

Note: The fast channel LIVETIME should only be used to determine the input count rate according to Equation 4-11. Specifically, it is NOT related to the energy filter livetime and should not be interpreted as the inverse of the processor deadtime. The DSP *does* calculate the energy filter livetime ELIVETIME, however, it is only an approximation. The most accurate deadtime measurement is obtained from ICR_m and OCR in Equation 4-11, as discussed in §4.11

4.10 Throughput

Figure 4.12 shows how the values of ICR_m and OCR vary with true input count rate for the DXP and compare these results to those from a common analog shaping amplifier plus SCA system. The data were taken at a synchrotron source using a detector looking at a CuO target illuminated by x-rays slightly above the Cu K absorption edge. Intensity was varied by adjusting two pairs of crossed slits in front of the input x-ray beam so that the harmonic content of the x-ray beam striking the detector remained constant with varying intensity.

NOTE: *The DXP's peaking time is* ***twice*** *as long as the analog system peaking time in this comparison, and yet the throughput is nearly the same.*

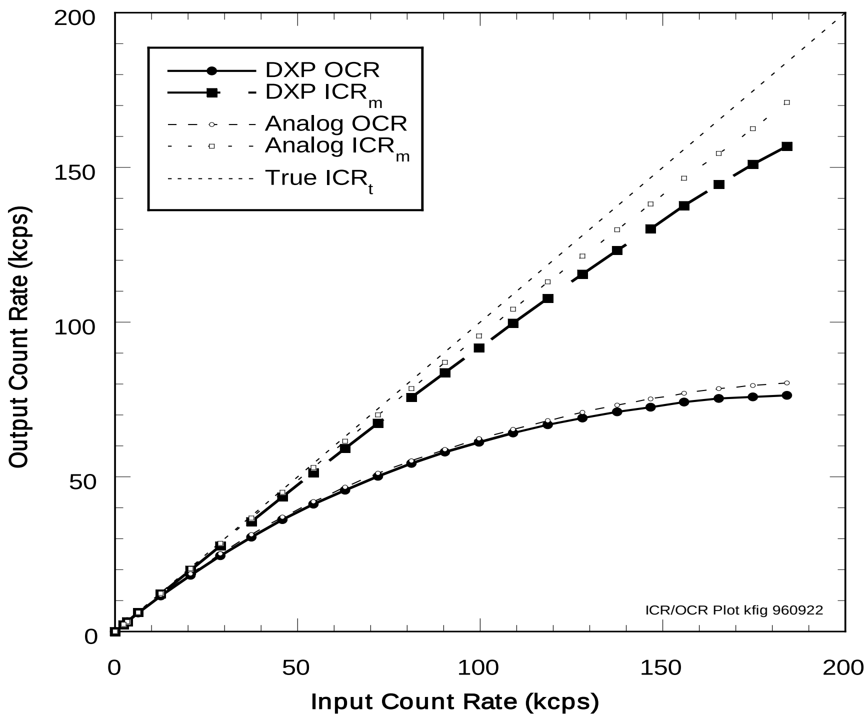

Figure 4.12: Curves of ICR_m and OCR for the DXP using 2 µs peaking time, compared to a common analog SCA system using 1 µs peaking time.

System	OCR Deadtime (µs)	ICR Deadtime (µs)
DXP (**2 µs** τ_p, 0.6 µs τ_g)	4.73	0.83
Analog Triangular Filter Amp (τ_p = **1 µs**)	4.47	0.40

Table 4.3: Comparing the deadtime per event for the DXP and an analog shaping amplifier. Notice that that the DXP produces a comparable output count rate even though its peaking time is nearly twice as long.

Functionally, the OCR in both cases is seen to initially rise with increasing ICR and then saturate at higher ICR levels. The theoretical form, from Poisson statistics, for a channel that suffers from paralyzable (extending) dead time is given by:

$$OCR = ICR_t * \exp(- ICR_t * \tau_d),$$

Equation 4-12

where τ_d is the *dead time*. Both the DXP and analog systems' OCRs are so describable, with the *slow channel dead times* - τ_d - shown in Table 4.3. The measured ICR_m values for both the DXP and analog systems are similarly describable, with the *fast channel dead times* - τ_{df} - as shown. The maximum value of OCR can be found by differentiating Equation 4-12 and setting the result to zero. This occurs when the value of the exponent is -1, i.e. when ICR_t equals $1/\tau_d$. At this point, the maximum OCR_{max} is 1/e the ICR, or

$$OCR_{max} = 1/(e\ \tau_d) = 0.37/\tau_d$$

Equation 4-13

These are general results and are very useful for estimating experimental data rates.

Table 4.3 illustrates a very important result for using the DXP: the slow channel deadtime is nearly the minimum value that is theoretically possible, namely the pulse basewidth. For the shown example, the basewidth is 4.6 μs ($2L_s + G_s$) while the deadtime is 4.73 μs. The slight increase is because, as noted above, PEAKINT is always set slightly longer than L_s - $G_s/2$ to assure that pileup does not distort collected values of V_x.

The deadtime for the analog system, on the other hand is much larger. In fact, as shown, the throughput for the digital system is almost twice as high, since it attains the same throughput for a 2 μs peaking time as the analog system achieves for a 1 μs peaking time. The slower analog rate arises, as noted earlier both from the longer tails on the pulses from the analog triangular filter and on additional deadtime introduced by the operation of the SCA. In spectroscopy applications where the system can be profitably run at close to maximum throughput, then, a single DXP channel will then effectively count as rapidly as two analog channels.

4.11 Dead Time Corrections

The fact that both OCR and ICR_m are describable by Equation 4-12 makes it possible to correct DXP spectra quite accurately for deadtime effects. Because deadtime losses are energy independent, the measured counts N_{mi} in any spectral channel i are related to the true number N_{ti} which would have been collected in the same channel i in the absence of deadtime effects by:

$$N_{ti} = N_{mi}\ ICR_t/OCR$$

Equation 4-14

Looking at Figure 4.12, it is clear that a first order correction can be made by using ICR_m in Equation 4-11 instead of ICR_t, particularly for OCR values less than about 50% of the maximum OCR value. For a more accurate correction, the fast channel deadtime τ_{df} should be measured from a fit to the equation:

$$ICR_m = ICT_t * \exp(-ICR_t\ \tau_{df})$$

Equation 4-15

Then, for each recorded spectrum, the associated value of ICR_m is noted and Equation 4-15 inverted (there are simple numerical routines to do this for transcendental equations) to obtain ICR_t. Then the spectrum can be corrected on a channel-by-channel basis using Equation 4-12. In experiments with a DXP prototype, we found that, for a 4 μs peaking time (for which the maximum ICR is 125 kcps), we could correct the area of a reference peak to better than 0.5% between 1 and 120 kcps.

5 microDXP DSP Code Description

NOTICE: if you are curious about how the DSP operates in controlling the DXP and processing data from the FiPPI, then please read on. You will also find this information useful if you wish to develop your own control code for the microDXP. However, in the latter case, we strongly advise you to use XIA's support libraries (Handel) to interface between your program and the microDXP module.

5.1 Introduction and Program Overview

The following sections are intended to provide the DXP user with a good understanding of the various tasks performed by the DSP in the microDXP. The DSP performs several functions:

1. Respond to input and output calls from the host computer to start and stop data collection runs, download control parameters, and download collected data.
2. Perform system calibration measurements by varying the various DAC settings under its control and noting the output change at the ADC.
3. Make initial measurements of the slow filter baseline and preamplifier slope value at the start of data taking runs to assure optimum starting parameter values.
4. Collect data:
    a. Read energy values E_X from the FiPPI, under interrupt control, and store them in DSP buffer memory in less than 0.25 μs.
    b. Adjust the ASC control parameters, under interrupt control, to maintain its output within the ADC's input range.
    c. Process captured E_X values to build the x-ray spectrum in DSP memory.
    d. Sample the FiPPI slow filter baseline and build a spectrum of its values in order to compute the baseline offset for E_X values.

Several DSP program variants are available to cover a range of applications. The standard program provided with the microDXP is for typical x-ray fluorescence spectroscopy using a reset-type preamplifier. Additional program variants are available for other applications, including hardware diagnostics and other specialized measurements, e.g.:

- *X-ray mapping*
- *Quick XAFS scanning*
- *Switching between multiple spectra synchronously with an experimentally derived signal (e.g. "Phased locked EXAFS")*
- *Time resolved spectroscopy (e.g. "multi-channel scaling")*

Standard variants available to all users via our website are described in §5.11. Several other variants have been developed for particular customers and may be made available upon request.

By convention, the DSP programs are named “NAMEmmnn.HEX”, where NAME is the variant name listed in the table, mm and nn are major and minor version numbers, respectively. The hex file format is in ASCII, with the parameter table at the top followed by the code generated by the Analog Devices 218x development system.

The internal data memory area is subdivided into three sections. The first section, starting at location 0x4000, contains DSP parameters and constants, both those used for controlling the DSP's actions and those produced by the DSP during normal running. These parameters and their addresses are listed and described in the following sections. When these parameters are referred to they will be denoted by all capital letters (e.g. RUNTASKS). The locations of parameters can (and, for forward compatibility should) be determined from the symbol table.

The second section of data memory contains acquired monitoring data such as the baseline event histogram. The third section of internal data memory is used as a circular buffer for storing events from the FiPPI. Note that future hardware revisions may eliminate the need for this buffer area, in which case it could be switched to more histogramming area.

5.2 Program Flow

The flow of the DSP program is illustrated in Figure 5.1. It is essentially identical for all program variants. The structure is very simple; after initialization, the DSP enters an idle phase, waiting for a signal from the host to start a run. During this idle phase, the DSP is continuously collecting baseline events from the FiPPI as well as monitoring the Analog Signal Conditioner (ASC) to keep the ADC input signal in the proper range and to adjust the slope generator to match the current input rate. When the Begin Run signal is received (from the host through the CSR register), the DSP first determines whether the run is a normal data acquisition run or a special run.

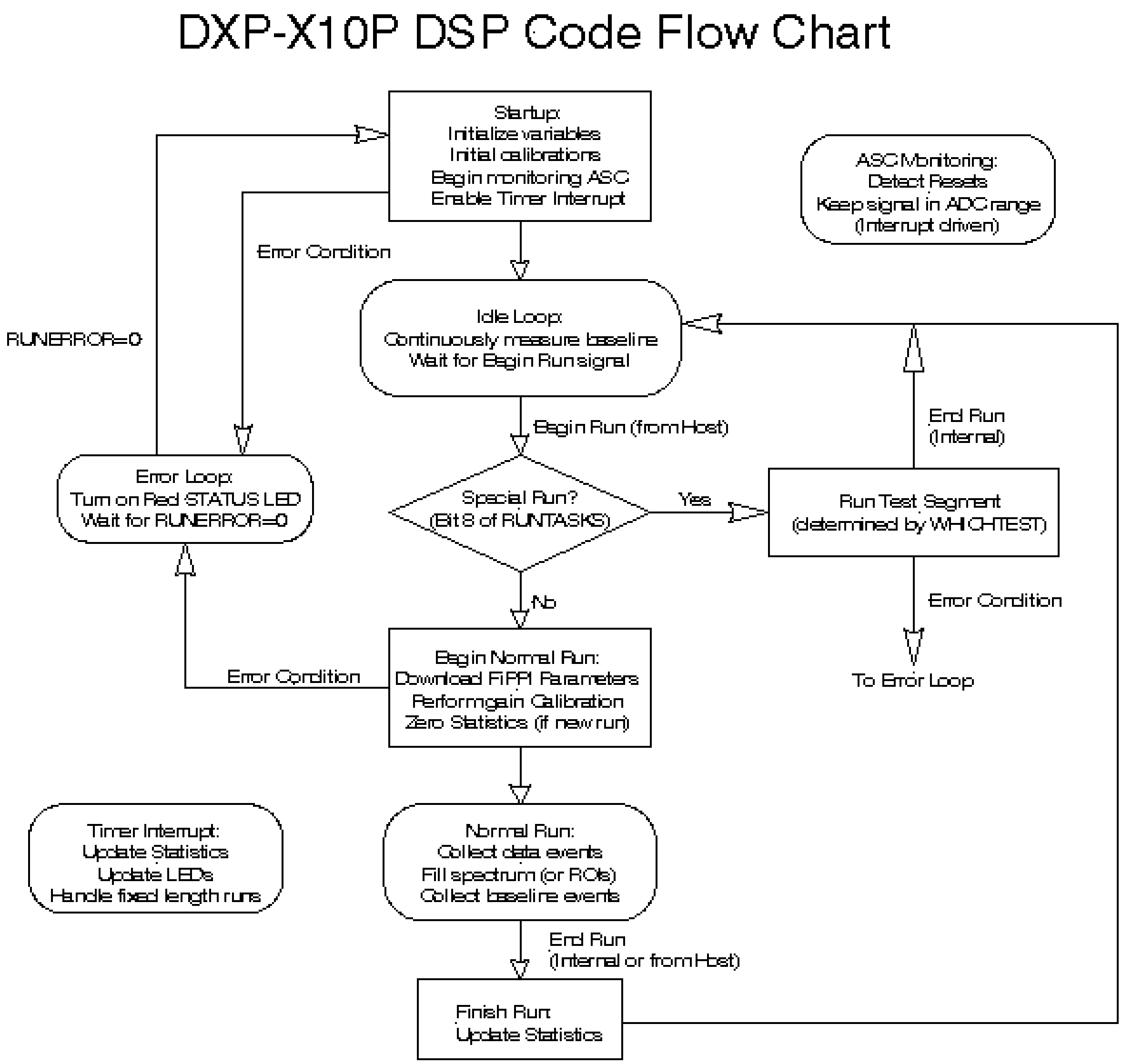

Figure 5.1 DSP code flow diagram.

In a normal run, ASC monitoring and baseline collection continue as in the idle phase. Event interrupts are enabled; when the FiPPI detects an event, it interrupts the DSP, which quickly responds and reads the energy value from the FiPPI into an internal buffer in data memory. The events in the buffer are then used to build the x-ray spectrum (or fill regions of interest).

In a special run, the action is determined by the value of the parameter WHICHTEST. The special runs include calibration tasks such as collecting an ADC trace, as well as ways of putting the DSP code into a special state (such as putting it into a dormant state to allow reprogramming the FiPPI on the fly). Special runs normally end automatically and the DSP returns to the idle state.

After the initialization phase, the Timer interrupt is enabled. This interrupt is used to handle the housekeeping type chores, such as updating the statistics during a run, controlling the rate LED, and handling fixed length runs. The Timer interrupt occurs with a period of 500 μsec.

If the DSP encounters an error condition, the DSP turns on the red status LED and waits for the host to set the parameter RUNERROR to 0 (after finding and fixing the problem that resulted in the error condition).

Each phase of the DSP program is discussed in more detail below.

5.3 Initialization

The DSP code starts running immediately after the power is turned on. During the initialization phase, several tasks are performed:

1. Setup internal DSP control registers
2. Zero spectrum and data memory, then initialize parameters to default values.
3. Set ASC DACs to initial default values
4. Initialize FiPPI and download default filter parameters
5. Perform initial calibrations for controlling the ASC:
   a. Find the SlopeDAC setting corresponding to zero slope
   b. Measure conversion factor used to calculate the contribution of the slope generator to the FiPPI baseline.
6. Enable the ASC and timer interrupts.

After the interrupts are enabled, the DSP is alive and ready to take data. After completing the initialization phase, the DSP enters the idle phase. In the idle phase, the DSP continuously samples the FiPPI baseline and updates the baseline subtraction register in the FiPPI so that the FiPPI is always ready to take data as soon as a run is started. There are two primary tasks performed during a normal data acquisition run: event processing and baseline measurement. These tasks are described in detail below.

5.4 Event Processing

5.4.1 Run Start

Prior to the start of a normal the run, the DSP performs several tasks:

1. Sets the stored gain (by setting the GAINDAC).
7. Sets the stored polarity.
8. Downloads FiPPI parameters (SLOWLEN, SLOWGAP, etc) stored in the current PARSET to obtain the desired peaking time.
9. Updates the internal calibrations with the new gain and FiPPI values.
10. If desired, the run statistics and the MCA are cleared. Otherwise, the run is treated as a continuation of the previous run. Note that for a run continuation, no gain or FiPPI changes are performed. In either case, the run number (parameter RUNIDENT) is incremented.

5.4.2 Event Interrupt

When the FiPPI detects a good event, it triggers a high priority interrupt in the DSP. Upon receiving the interrupt, the DSP immediately reads the event energy from the FiPPI into an internal circular buffer and increments the write pointer into that buffer. The normal event loop compares the write pointer to the read pointer to determine that there is a new event to process.

5.4.3 Event Loop

The processing that takes place during a normal collection run is very simple, in order to allow high event rates. The structure of the event loop is illustrated below in pseudocode:

```
while (RunInProgress)
      {
      if (EventToProcess)
            ProcessEvent
      else
            CollectBaseline
      endif
      }
RunFinish
goto IdleLoop
```

The run can be stopped by the host, or can be stopped internally for fixed length runs; see §5.10.1 below.

The event processing involves either binning the energy into an MCA or determining whether the event falls into a defined SCA window, depending upon the DSP code variant. If there is no event to process, the DSP reads a baseline value from the FiPPI; see below for a detailed description of the baseline processing. Once the run is over, the statistics are finalized and the DSP returns to the idle state where it continuously samples baseline and waits for a command to start a new run.

5.4.4 Spectrum Binning

The primary event-processing task is to use the energies measured in the FiPPI to build up a full energy spectrum (MCA). The MCA bin width is determined by the analog gain, the FiPPI filter length and the binning parameter BINFACT1. The DSP determines the spectrum bin by multiplying the FiPPI energy output by (1/BINFACT1). If the bin is outside the range determined by the parameters MCALIMLO and MCALIMHI, the event is classified as an underflow or overflow. Otherwise, the appropriate bin is incremented. A 24-bit word is used to store the contents of each bin, allowing nearly 16.8 million events per MCA channel.

5.4.5 SCA Mapping

An alternate variant of the DSP code allows the user to define up to 24 SCA regions and count the number of events that fall into each region. The regions are defined in terms of MCA bin number, and can overlap. A useful method for defining the SCA windows is to take a run with the full MCA

spectrum, and use the spectrum as an aid in choosing the limits for each SCA. The reduced amount of data storage in SCA mapping mode is very useful in time resolved spectroscopy or scanning applications, where separate spectral data are desired for many different time or spatial points.

5.5 Baseline Measurement

The DSP collects baseline data from the FiPPI whenever there are no events to process, both during a run and between runs (when there are never events to process). The DSP keeps a running average of the most recent baseline samples; this average is written back into the FiPPI where it is subtracted from the raw energy filter value to get the true energy. The baseline data read from the FiPPI is just the raw output of the energy filter. One bit of the baseline register is used to indicate whether the sample occurred while an event was in progress, in which case it is not used.

Two methods are available to determine the average baseline value. By default, an *infinite impulse response (IIR)* filter is used, where the baseline average is calculated by combining a new baseline sample with the old average, using weights x and (1-x) respectively, where x is typically 1/128. By setting the appropriate bit in the parameter RUNTASKS (see below), a *finite impulse response (FIR)* filter is used, where the baseline mean is just the straight average of the N most recent baseline samples. Both averaging methods are described in more detail in the following sections. The baseline mean is stored with 32 bit precision in the parameters BASEMEAN0 (high order word) and BASEMEAN1.

5.5.1 IIR (Infinite Impulse Response) Filter

By default, the baseline mean is calculated using an infinite impulse response filter, characterized in the following way:

$$< B_i >= \frac{N-1}{N} < B_{i-1} > + \frac{1}{N} B_i$$

Equation 5-1

Where $<B_i>$ is the baseline mean after the i^{th} baseline sample, B_i is the i^{th} baseline sample, and $<B_{i-1}>$ is the baseline mean before the i^{th} sample. With this filter, the most recent baseline samples are weighted the most, but (up to the precision of the stored mean value) all baseline values have a small effect on the mean (hence the infinite in the name).

The length of the filter is controlled by the parameter BLFILTER, which holds the value 1/N in 16 bit fixed point notation, which has 1 sign bit and 15 binary bits to the right of the decimal point. Expressed as a positive integer, BLFILTER = (1/N)*2**15. The default value for BLFILTER corresponds to N=64. Interpreting BLFILTER as an integer gives (1/128)*2**15 = 2**9 = 256.

5.5.2 FIR (Finite Impulse Response) Filter

By setting the appropriate RUNTASKS bit, it is possible to choose a finite impulse filter to calculate the baseline mean. With this filter, a straight average of the N most recent valid baseline samples is used to calculate the mean. To implement this filter, a buffer large enough to hold all N samples is necessary. For this reason, the length of the finite response filter is limited to 1024. The filter length is stored in the parameter BLFILTERF.

5.5.3 Baseline Histogram

As part of the baseline processing, all valid baseline samples are entered into the baseline histogram, which occupies 1024 words of data memory. The baseline histogram can be very useful in monitoring or evaluating the performance of the microDXP, as discussed in the **Rapid Development Kit Manual**. The parameter BASESTART contains the pointer to the location of the histogram in data memory, and the length (nominally 1K) is contained in the parameter BASELEN.

The baseline histogram is centered about a zero baseline. The parameter BASEBINNING determines the granularity of the histogram; 2**BASEBINNING baseline values are combined into one bin of the baseline histogram. The default value of BASEBINNING is 2 (i.e., the baseline value is divided by 4 to determine the bin). All valid baseline values are included in the histogram, even if there is a baseline cut (see §5.5.5 below) in use.

The baseline histogram is only filled during a normal data acquisition run; when the DSP is idle, the baseline average is calculated but the histogram is not filled. Since the baseline histogram is stored in data memory, 16-bit words are used to record the bin contents. As a result, the histogram overflows quite often; the time to overflow depends on the baseline sample rate (typically several 100 kHz) and the width of the baseline distribution. When the DSP detects an overflow, all bins are scaled down by a factor of 2 and histogramming continues.

The baseline distribution should be very Gaussian; the width of the distribution reflects the electronic noise in the system (including the effects of the energy filter). A tail on the positive side of the distribution indicates the presence of energy in the baseline, resulting from undetected pileup or energy depositions that did not satisfy the trigger threshold. The tail should be very small compared to the peak of the histogram; it will grow with rate. If this tail is too large, it can have a noticeable effect on the baseline mean, leading to negative peak shifts. Under these circumstances, enabling the baseline cut is useful in eliminating the bias.

A tail on the low energy side of the baseline distribution is usually caused by baseline samples just after a preamplifier reset; the effects of the reset can last quite a while (tens of microseconds), especially for optical reset preamplifiers. It is usually best not to take data while the reset is in effect; the dead time associated with a reset can be adjusted using the parameter RESETWAIT, which sets the dead time in units of 250.

5.5.4 Residual Baseline

When operating with a reset type preamplifier, the raw baseline measured in the FiPPI (which is just the output of the energy filter) comes from two sources: the detector preamplifier and the slope generator in the microDXP itself. At high rates, the slope gets rather large in order to balance the high-energy deposition rate in the detector; under these conditions, the baseline due to the slope is by far the dominant factor in the baseline.

By default, the DSP continually adjusts the slope to match the current rate; these slope adjustments result in an instantaneous change in the baseline. If the baseline due to the slope generator is included in the baseline mean, the change in the calculated mean would be delayed relative to the change in the slope, due to the effect of all the baseline samples prior to the slope change. For this reason, the baseline due to the slope is subtracted out of the overall baseline prior to calculating the mean value (and added back in prior to loading the FiPPI baseline subtraction register). The *residual* baseline included in the mean reflects the detector leakage current, and should be fairly constant with rate (to the extent that the leakage current does not depend on rate). The calibration procedure used to determine the baseline due to the slope generator is performed during the initial startup procedure.

By default, the baseline due to the slope generator is taken out of the baseline average. You can choose to include the slope baseline in the mean (e.g. for diagnostic purposes) you can do so by clearing the residual baseline bit (6) in RUNTASKS.

5.5.5 Baseline Cut

As specified above, a baseline cut is available to exclude baseline samples that include real event energy, which can lead to peak shifting at high event rates. The cut is expressed as a fraction of the peak value of the baseline distribution; by default, the baseline cut is set to 5%. The cut values are based on the baseline histogram, and are recalculated every time the histogram overflows (every few seconds). The DSP searches on either side of the peak of the baseline distribution for the first bin whose contents are less than the cut (.05 by default) times the peak value; these bin numbers are used to calculate the actual baseline cut.

The cut fraction is stored in the parameter BLCUT, expressed in 16-bit fixed-point notation. Interpreted as an integer, BLCUT = (cut fraction)*2**15; the default 5% cut corresponds to BLCUT=1638 decimal (or 666 hex). The actual cut values determined by the DSP code are stored in BLMIN and BLMAX. The baseline cut is enabled or disabled by setting or clearing a bit (10) in the RUNTASKS parameter.

5.6 Interrupt Routines

There are several tasks performed under interrupt control within the DSP on the microDXP. The event interrupt routine (which just transfers event data from the FiPPI to an internal buffer) is described above in Section 5.4 above. There are two other interrupt routines: the ASC interrupt is used to keep the analog signal within the input range of the ADC, and the timer interrupt is

used to handle such housekeeping chores as updating statistics. These routines are described in more detail below.

5.6.1 ASC Monitoring

There are four main tasks performed by the ASC interrupt routine:

1. Detects Resets (reset-type detectors only)
2. Adjusts the slope generator to match the event rate (reset-type detectors only)
3. Adjusts the offset value to keep the signal in range (RC feedback detectors only)
4. Moves the signal back to the center of the ADC range whenever it drifts out of range (high or low).

The ASC interrupt routine is triggered whenever the FiPPI detects the ADC going out of range. The DSP first triggers a short reset. If the signal merely drifted out of range to begin with, the short reset will bring the signal back to the center of the ADC range, and data taking resumes. If instead the signal was driven out of range by a preamplifier reset, the short reset will not bring the signal back into range. In this case the DSP triggers a long reset, and then holds the signal at the center of the ADC range for a time determined by the parameter RESETINT, which specifies the dead time after a reset in microsecond units. After the reset interval, the signal is released and data taking resumes.

The DSP keeps track of how many times the signal drifts out of range in both directions, and adjusts the slope such that the number of drifts high (NUMDRUPS) roughly matches the number of drifts low (NUMDRDOS). If the DSP determines that the slope must be changed to match the rate, the SlopeDAC value is modified by a constant fraction of the parameter SLOPEVAL determined by the value of the parameter SGRANULAR. By default, the slope adjustment granularity is 5%, which is a good compromise between adjusting the slope quickly to match quickly changing input rates and being able to set the SlopeDAC just right.

For an RC feedback detector, the offset added to the input signal is adjusted such that the signal stays in range as much as possible.

5.6.2 Timer Interrupt

Every 500 μsec, the DSP is interrupted to take care of the regular 'maintenance' type tasks. These tasks include:

1. Update the run statistics EVTSINRUN, LIVETIME, REALTIME and FASTPEAKS (only during a run).
2. Control the Rate LED. This LED flashes whenever a reset is detected (reset detector only), and during a run the color indicates the current output/input ratio. By default, the LED flashes green for OCR/ICR>0.5, flashes yellow (green plus red) for 0.5>OCR/ICR>1/e, and flashes red for OCR/ICR<1/e. The thresholds are determined by the parameters YELTHR and REDTHR.

3. Handle fixed length runs. During a fixed length run, the current value of EVTSINRUN (output events), FASTPEAKS (input events), LIVETIME or REALTIME is compared to the desired run length. Once the value exceeds the desired value, the run is ended.

5.7 Error Handling

When the DSP detects an error in the operation of the microDXP, the red Status LED is turned on, and the source of the error is stored in the parameter RUNERROR. The possible values for RUNERROR are listed below:

RUNERROR Value	Meaning
0	No Error
1	FiPPI communication error
2	ASC setup failure
3-5	Reserved
6	TrackDAC calibration error

Table 5.1: Identification of DXP errors according to the DSP parameter RUNERROR.

A FiPPI communication error could mean that the FiPPI configuration was not successful. An ASC calibration error can indicate a hardware problem, or possibly that a jumper is not set properly (for example, the DSP code for reset preamplifiers will generate an error if the jumper is set to run in OFFSET mode).

Once the source of the error has been located and cleared, the host can set RUNERROR to 0 to force the DSP to exit the error loop and reinitialize the system. Note that all system settings are saved when initialization is performed coming out of the error loop. Of course, another valid method for clearing the error is to re-download the DSP code after fixing the problem.

5.8 Specifying Data Acquisition Tasks (RUNTASKS):

Many aspects of the operation of the microDXP are controlled by individual flag bits of the parameter RUNTASKS. The meaning of each RUNTASKS bit is described below:

Bit	Meaning if set (1)	Meaning if cleared (0)
0	Reserved (set to 0)	Reserved (set to 0)
1	Update SlopeDAC or OffsetDAC value to match current rate (DEFAULT)	SlopeDAC or OffsetDAC adjustments disabled
2	Use Finite Impulse Response (FIR) filter to calculate baseline average	Use Infinite Impulse Response (IIR) filter to calculate baseline average (DEFAULT)
3	Acquire baseline values for histogramming and averaging (DEFAULT)	Disable baseline acquisition
4	Adjust fast filter threshold to compensate for rate shifts	Disable fast filter threshold adjustment
5	Correct for baseline shift, either in FiPPI (pulse reset) or DSP (RC feedback) (DEFAULT)	Disable baseline correction
6	Apply residual baseline correction (DEFAULT)	No residual baseline correction
7	Disable writing baseline values to baseline history circular buffer	Continuously write baseline values to baseline history circular buffer (DEFAULT)
8	Indicates special task or calibration run specified by WHICHTEST	Indicates normal acquisition run

9	Histogram DeltaBaseline (baseline - <baseline>)	Histogram raw baseline (DEFAULT)
10	Enable baseline cut	Disable baseline cut (DEFAULT)
11-15	Reserved (set to 0)	Reserved (set to 0)

Table 5.2: Data acquisition tasks controlled by the DSP parameter RUNTASKS.

5.9 Special Tasks (WHICHTEST)

Special tasks are selected by starting a run with bit 8 of the RUNTASKS parameter set. The following tasks are currently supported:

Number	Test Segment
0	Set ASC DAC values to current value of GAINDAC, SLOPEDAC or OFFSETDAC
1	Acquire ADC trace in history buffer
2	Gain calibration
3	Slope calibration (measure SLOPEMULT)
4	Measure ADC non-linearity
5	Not currently used
6	Put DSP to sleep while FPGA logic is downloaded
7	Not currently used
8	OffsetDAC calibration (measure OFFDACVAL)
9-10	Not currently used
11	Program FiPPI
12	Set internal polarity to current value of POLARITY parameter
13	Not currently used
14	Not currently used
15	RC feedback calibration trace of baseline filter and decimator values
16	RC feedback calibration trace of event filter and decimator values

Table 5.3: Special tasks and test segments that can be selected with the DSP parameter WHICHTEST.

5.10 DSP Parameter Descriptions

As noted above, DSP operation is based on a number of parameters. Some are control parameters required to operate the DXP, some are calibration values determined by the DSP, and others are run statistics.

NOTE: in general you will not want to modify these parameters directly, but only through a host control program like microManager or, if you are a programmer, through a software library like XIA's Handel library.

Variable	***Type***	***Description***	***Reference***
PROGNUM	Constant	Program variant number.	
CODEREV	Constant	Current DSP program revision.	
HDWRVAR	Constant	Hardware variant. DSP reads this from interface FPGA.	
FIPPIREV	Constant	FiPPI design revision. DSP reads this from FiPPI FPGA.	
FIPPIVAR	Constant	FiPPI design variant. DSP reads this from FiPPI FPGA.	
DECIMATION	Constant	Slow filter decimation factor. DSP reads this from FiPPI FPGA.	
RUNIDENT	Returned	Run identifier	
RUNERROR	Returned	Error code if run is aborted, 0 for success	
BUSY	Returned	DSPs current acquisition status. Values listed below.	
<u>Acquisition Statistics:</u>			
LIVETIME0,1,2	Statistic	Intermediate filter live time in 500 nsec units	

ELIVETIME0,1,2	Statistic	Energy filter live time in 500 nsec units
REALTIME0,1,2	Statistic	Elapsed acquisition time in 500 nsec units
EVTSINRUN0,1	Statistic	Number of events in MCA spectrum
UNDRFLOWS0,1	Statistic	Number of MCA underflow events
OVERFLOWS0,1	Statistic	Number of MCA overflow events
FASTPEAKS0,1	Statistic	Number of input events detected by FiPPI
NUMASCINT0,1	Statistic	Number of ASC interrupts
NUMRESETS0,1	Statistic	Number of "reset" events seen
NUMUPSETS0,1	Statistic	Number of "upset" events seen
NUMDRUPS0,1	Statistic	Number of "drift up" events seen
NUMDRDOS0,1	Statistic	Number of "drift down" events seen.
NUMZIGZAG0,1	Statistic	Number of "zigzag" events seen
BASEEVTS0,1	Statistic	Number of baseline events acquired
BASEMEAN0,1	Statistic	Updating mean baseline value
Control parameters:		
WHICHTEST	Parameter	Which test segment to execute.
RUNTASKS	Parameter	Which tasks will be executed in run sequence
BINFACT1	Parameter	MCA binning factor
MCALIMLO	Parameter	Lower limit of MCA spectrum
MCALIMHI	Parameter	Upper limit of MCA spectrum
TRACEWAIT	Parameter	ADC trace time factor
ASCTIMOUT	Parameter	Timeout for ASCSetup in tenths of seconds
YELLOWTHR	Parameter	Medium rate throughput threshold for front panel LED
REDTHR	Parameter	High rate throughput threshold for front panel LED
PRESET	Parameter	Preset type (0:none; 1:real time; 2:live time; 3: output cts; 4: input cts)
PRESETLEN0,1	Parameter	Preset run length
FiPPI Digital Filter/Event selection parameters:		
SLOWLEN	Parameter	Slow filter length
SLOWGAP	Parameter	Slow filter gap
PEAKINT	Parameter	Peak interval
FASTLEN	Parameter	Fast filter length
FASTGAP	Parameter	Fast filter gap
THRESHOLD	Parameter	Threshold value for fast filter trigger (range: 1-255, 0 disables)
MINWIDTH	Parameter	Minimum peak width
MAXWIDTH	Parameter	Maximum peak width
BASETHRESH	Parameter	Automatically set threshold for intermediate filter trigger (range 1-255, 0 disables both baseline and event discrimination—use FIPCONTROL to disable event discrimination only)
BASETHRADJ	Parameter	Coefficient for BASETHRESH auto-set algorithm (range 1-255, smaller values result in tighter thresholds)
FIPCONTROL	Parameter	FiPPI advanced control, bitwise flag register: bit 0 – fast threshold (0: enabled) bit 1 – intermediate threshold (0: enabled) bit 2 – slow threshold (0: enabled)
SLOWTHRESH	Parameter	Threshold for slow filter trigger (range 1-255, 0 disables)
PEAKSAM	Parameter	Peak sampling time
Baseline Related Parameters:		
BLFILTER	Parameter	Filtering parameter for baseline (IIR filtering)
BLFILTERF	Parameter	Filtering parameter for baseline (FIR filtering)
BASEBINNING	Parameter	Baseline binning for histogram (0:finest to 6:coarsest)
BLCUT	Parameter	DSP baseline cut (cut at BLCUT*FWHM, units defined below)
BLMIN	Calibration	Min baseline value accepted in average (calculated from BLCUT)
BLMAX	Calibration	Max baseline value accepted in average (calculated from BLCUT)
ASC Control Parameters and Calibrations (all variants)		
POLARITY	Parameter	Preamplifier signal polarity (0:negative step; 1:positive step)

GAINDAC	Parameter	Current Gain DAC value (16 bit serial DAC, range 0-65535).
ASC Control Parameters and Calibrations (reset-type variants)		
RESETWAIT	Parameter	Quick Reset time, 25ns units
RESETINT	Parameter	Reset time, 0.25 μsec units
SLOPEDAC	Calibration	Current Slope DAC value (16 bit serial DAC, range 0-65535)
SLOPEZERO	Calibration	Slope DAC zero value (approximately center of range)
SLOPEVAL	Calibration	Absolute setting (SLOPEDAC-SLOPEZERO)
SGRANULAR	Parameter	Slope DAC step size
TRKDACVAL	Parameter	Tracking DAC value: 12-bit parallel
TDACWIDTH	Parameter	Track DAC pulse width 50 ns units
TDQPERADC	Calibration	
TDQPERADCE	Calibration	
ASC Control Parameters and Calibrations (RC feedback variants)		
OFFSETDAC	Parameter	Current offset DAC value (16 bit serial DAC, range 0-65535).
OFFSETSTEP	Parameter	Offset DAC step size
TAURC	Parameter	Preamplifier decay constant, in 25 ns units (RCF variant only)
RCFCOR	Calibration	Preamplifier decay correction (RCF variant only)
Miscellaneous Constants:		
SPECTSTART	Constant	Address of MCA spectrum in program memory
SPECTLEN	Constant	Length of MCA spectrum buffer
BASESTART	Constant	Address of baseline histogram in data memory (offset by 0x4000)
BASELEN	Constant	Length of baseline histogram
EVTBSTART	Constant	Address of event buffer in data memory (offset by 0x4000)
EVTBLEN	Constant	Length of baseline histogram
HSTSTART	Constant	Address of history buffer in data memory (offset by 0x4000)
HSTLEN	Constant	Length of history buffer
NUMSCA	Parameter	Number of SCA regions defined (mapping variants only)
SCAxLO, x=0-23	Parameter	Lower MCA channel for SCA region x (mapping variants only)
SCAxHI, x=0-23	Parameter	Upper MCA channel for SCA region x (mapping variants only)
USER1-USER8	User	User variables. Host software can use these for any purposes

Table 5.4: Summary of DSP parameter definitions

5.10.1 Specifying fixed run lengths

`(PRESET, PRESETLEN0,1)`

By default, the microDXP acquires data until a stop command is received from the host.

A fixed run length can be specified using the parameters PRESET and PRESETLEN0,1, as follows:

PRESET specifies the type of run: 0 = indefinite (default)
1 = fixed realtime
2 = fixed (energy filter) livetime
3 = fixed output events
4 = fixed input counts

PRESETLEN0,PRESETLEN1 specifies the length of preset fixed length run, as a 32 bit quantity. For fixed real time or live time, the units are 500 nanosecond intervals.

5.10.2 Setting the slow filter parameters

`(SLOWLEN, SLOWGAP)`

The DXP uses a trapezoidal filter, characterized by the peaking time, T_p, and gap time, T_g. The peaking time is determined by the SLOWLEN and DECIMATION values. SLOWLEN is the interval of time, in units of decimated clock cycles, during which the decimated ADC signal is integrated, referred to as the peaking time. DECIMATION is automatically sensed by the DSP and should not be modified. For T_p and T_g. in µsec, and the pipeline clock F_{CLK} in MHz, the following gives the value of SLOWLEN and SLOWGAP:

$$\mathrm{SLOWLEN} = F_{CLK} * T_p * 2^{-\mathrm{DECIMATION}}$$

e.g. At DECIMATION = 4, T_p = 16 µsec yields SLOWLEN=8 for an 8MHz clock, or SLOWLEN=16 for a 16MHz clock. The user will want to be able to choose the peaking time based on resolution and throughput requirements as described earlier in this document.

SLOWGAP is the gap time, visible as the 'flat-top' region of the trapezoid.

$$\mathrm{SLOWGAP} = F_{CLK} * T_g * 2^{-\mathrm{DECIMATION}}$$

Subject to the restriction that it must exceed 3, SLOWGAP is set such that the flat-top interval is longer than the 0%-100% risetime of the preamplifier output pulses by at least 1 decimation period:

$$\text{Gap Time} = (2^{D} * \mathrm{SLOWGAP} * 125 \text{ ns}) > \text{pulse risetime} + 2^{D} * 125 \text{ ns}$$

$$\text{or: } (2^{D} * (\mathrm{SLOWGAP}-1) * 125 \text{ ns}) > \text{pulse risetime}$$

5.10.3 Setting the fast filter parameters

`(FASTLEN, FASTGAP)`

The fast filter is also trapezoidal but has a decimation of 0 for all FiPPI designs. The values of FASTLEN and FASTGAP are given, for T_p' fast peaking time and T_g' = fast gap time in µsec:

$$\mathrm{FASTLEN} = 8 * T_p'$$

$$\mathrm{FASTGAP} = 8 * T_g'$$

Typical values of these parameters are FASTLEN=4 and FASTGAP=0.

5.10.4 Setting Thresholds

`(THRESHOLD, MINWIDTH, BASETHRESH, BASETHRADJ, SLOWTHRESH, FIPCONTROL)`

Note: BASETHRESH is NOT used in the decimation 0 FiPPI (i.e. peaking time range 0.75 – 3 µsec for the standard 8MHz pipeline clock; or peaking time range 0.375 – 1.5 µsec for a 16MHz pipeline clock). Instead THRESHOLD is used to discriminate both for and against events.

X-rays are identified when a filter output goes above an active threshold. Thresholds can be applied to the fast (THRESHOLD), intermediate (BASETHRESH, BASETHRADJ) and energy filters (SLOWTHRESH), though

in practice the energy threshold is rarely used. The user will typically only adjust the fast filter threshold, THRESHOLD.

All thresholds are scaled by their respective filter lengths. Further, they cannot be expressed in energy units until the DXP conversion gain (see §5.10.6 below), G_{DXP} = number of ADC counts per keV at the DXP input, is known. For an energy threshold E_{th} in keV,

$$THRESHOLD=G_{DXP}*E_{th}*FASTLEN$$

$$BASETHRESH=G_{DXP}*E_{th}*SLOWLEN$$

$$SLOWTHRES=G_{DXP}*E_{th}*SLOWLEN$$

THRESHOLD is set by the user. A good procedure is to initially set the value too high. Once a spectrum is observed, reduce THRESHOLD until the zero energy noise peak starts to become significant, and then raise it again until only a trace of the noise peak remains. The MINWIDTH parameter is used for noise rejection: It is the minimum number of time bins the fast filter is above threshold. A typical value that works with FASTLEN=4 is MINWIDTH=4.

Currently BASETHRESH must also be set by the user (BASETHRESH will be automatically set by the DSP in the next release) and applied to the intermediate filter as part of the baseline acquisition circuitry, i.e. baseline measurements are taken when the signal is below this threshold. BASETHRESH threshold crossings by default also trigger event processing, effectively extending the detectable energy range significantly below the fast filter THRESHOLD. The parameter BASETHRADJ controls the algorithm that calculates BASETHRESH, with larger BASETHRADJ values resulting in more conservative BASETHRESH values. In the rare case the baseline threshold is set incorrectly by the DSP algorithm, we recommend adjusting BASETHRADJ rather than BASETHRESH itself.

The use of a slow threshold introduces significant errors in the counting statistics. Specifically, the dead-time-per-event is markedly different for x-rays above and below the threshold. SLOWTHRESH should only be used if:

- ✓ Your detector has a very thin window and operates in a vacuum.
- ✓ You understand that it is not possible to compute input count rates for x-ray peaks below the threshold relative to x-ray peaks above the threshold.

Setting a threshold to zero disables that threshold. Individual thresholds can also be enabled and disabled via the lowest 3 bits of the parameter FIPCONTROL:

FIPCONTROL BIT	Meaning, if = 0	Meaning, if = 1
0	THRESHOLD event discrimination *enabled*	THRESHOLD event discrimination *disabled*
1	BASETHRESH event discrimination *enabled*; baseline discrimination *enabled*	BASETHRESH event discrimination *disabled*; baseline discrimination ***enabled***
2	SLOWTHRESH event discrimination *enabled*	SLOWTHRESH event discrimination *disabled*

Table 5.5: Threshold control via FIPCONTROL.

The two methods of disabling thresholds are equivalent, except in the case of the intermediate filter threshold BASETHRESH, which is used for both baseline and energy discrimination,

e.g. setting FIPCONTROL = XXXX XXXX XXXX X110 enables event discrimination based on the fast filter THRESHOLD, disables event discrimination based on the intermediate filter BASETHRESH and slow filter SLOWTHRESH, however, BASETHRESH still discriminates baseline measurements in the intermediate filter; alternatively, setting BASETHRESH to zero would disable both event and baseline discrimination, regardless of the value of FIPCONTROL.

5.10.5 Setting the Pile-up inspection parameters

(MAXWIDTH, PEAKINT)

MAXWIDTH is used to reject pulse pile-up on a time scale that is comparable to FASTLEN as discussed in § 4.8. A typical value is

MAXWIDTH = 2*FASTLEN + FASTGAP +N

where N is in the range 4-8. If the signal rise-time depends on the x-ray energy (e.g. bandwidth limited preamplifier or low field regions of the detector that are preferentially sampled at some energy) this cut can bias the spectrum if it is too small.

PEAKINT is used to reject energy channel pulse pile-up when the pulses are well resolved by the fast channel. This value should be set as:

PEAKINT = SLOWLEN + SLOWGAP + N

where N = 1 typically.

5.10.6 Setting the Analog Gain (GAINDAC)

The DXP internal gain is chosen to set the ADC dynamic range appropriately for the signals of interest. If it is set too low, the energy resolution may be compromised, while if is set too high there may be excessive deadtime and attenuation of higher energy x-rays. The ADC range is one (1) Volt full scale. Two guidelines are suggested for the internal gain setting:

1. This is appropriate when there is a single peak of interest: Set the gain such that the typical pulse height is between 2 and 10% of the ADC range.
2. This is appropriate when looking at a fixed energy range, with no particular peak of interest: Set the gain such that the maximum energy pulses are around 200 displayed vertical units in the ADC Trace readout.

The parameter GAINDAC sets the internal amplifier gain. The overall gain can be expressed as follows:

$$G_{tot} = G_{in}' * G_{var} * G_{base}$$

where

G_{in}': the nominal input stage gain G_{in} ~ 1, is modified by the resistive divider created by the output impedance of the preamplifier and the input impedance of the microDXP to the value G_{in}'.

G_{var}: variable gain setting = 0.36 to 36 depending on GAINDAC setting

G_{base}: reference gain ~ 1.87

The total internal gain ranges from 0.67 V_{ADC}/V_{INPUT} to 67.3 V_{ADC}/V_{INPUT}.

The digital gain control is a 16-bit DAC that sets the gain of a "linear in dB" variable gain amplifier. The gain setting accuracy is approximately one bit (or 0.00061 dB = 0.007%). The relationship between G_{var} and GAINDAC is:

$$\text{Gain (in dB)} = (\text{GAINDAC}/65536) * 40 \text{ dB}$$

$$G_{var} = 10^{**}(\text{Gain (in dB)}/20)$$

The output impedance R_{OUT} of the preamplifier creates a resistive divider with the microDXP input impedance R_{IN} =1.0kΩ and thus affects the input gain term G_{in}:

$$G_{in}\ (R_{OUT})' = 1.0k\Omega\ /(1.0k\Omega + R_{OUT}))$$

5.11 Standard Program Variants

5.11.1 MCA acquisition with reset-type preamplifiers

Variant 0 is the standard firmware variant supplied with the microDXP, as described in this manual. It is intended for use with reset-type preamplifiers (described in §4.1.1). *Note:* To use this variant, the "Ramp/Offset" switch S1 (see Appendix D) should be in the "Ramp" position.

5.11.2 MCA acquisition with RC-type preamplifiers

This firmware variant is intended for use with resistive feedback preamplifiers (described in §4.1.2). Additional parameters (described in §4.7):

TAURC: Exponential decay time in 50 ns units.

RCFCOR: Correction factor (calculated automatically at start of run if TAURC not 0)

Note: To use this variant, the “Ramp/Offset” switch S1 (see Appendix D) should be in the “Offset” position.

Appendices

Appendix A. GLOBSET Specification

ID	*Parameter*	*Format*	*Description*
0	NUMGLOBSET	16:0	Number of GLOBSET parameters NOT including NUMGLOBSET and GLOBVERSION (used internally)
1	GLOBVERSION	16:0	Version of the GENSET (used internally)
2	POLARITY	1:0	Preamplifier signal polarity (0: negative, 1: positive)
3	RUNTASKS	Bitwise flag register	Each bit controls a separate task. Will be in manual.
4	FIPCONTROL		Controls various FiPPI operations. Used for debugging primarily.
5	PRESETLENLO		Low word of preset run length
6	PRESETLENHI		High word of preset run length
7	PRESET		Preset type (0: no preset, 1: realtime, 2: livetime, 3: output events, 4: input events)
8	RESETINT		Reset time in microseconds
9	TAURC		Preamplifier RC decay time in microseconds (unused for reset preamplifier)
10	IDLEMODE		Idle mode, as defined in command 0x46
11	IDLEDELAY		Delay before entering idle after end run (in seconds)
12	SLEEPMODE		Sleep mode, see command 0x47
13	TRACEWAIT		Number of clock ticks between successive samples when taking ADC trace (see command 0x11)
14	STATSMODE		Sets the response to the 0x06 “Read Run Statistics” command. STATSMODE=0 (default) yields the short response (21 bytes). STATSMODE=1 yields the long response is (29 bytes, including UNDERFLOWS and OVERFLOWS).

Appendix B. GENSET Version 1 Specification

Five MCA related GENSETs (i.e. five MCA formats) are available.

ID	*Parameter*	*Format*	*Description*
0	NUMGENSET	16:0	Number of GENSET parameters NOT including NUMGENSET and GENVERSION (used internally)
1	GENVERSION	16:0	Version of the GENSET (used internally)
2	MCALEN		Number of spectrum bins
3	MCALIMLO		Lowest spectrum bin (allows offset spectrum)
4	MCALIMHI		Highest spectrum bin
5	BASEBINNING		Number of bins baseline values combined into one bin in baseline histogram (power of 2)
6	BLCUT		Baseline cut value, in percentage of peak value in baseline histogram. Standard value is 5%. Expressed as x * 32768.
7	BINMULTIPLE		MCA bin size in terms of the minimum.
8	BINGRANULAR		Bin granularity (as defined in command 0x83: 0 == very fine, etc)
9	GAINBASE		16-bit GAINDAC base setting. (Higher value == higher gain). This value is modified per-peaking time by the GAINTWEAK table in the PARSET.

Appendix C. PARSET Version 1 Specification

These are all the parameters that are stored and retrieved for each peaking time.

ID	*Parameter*	*Format*	*Description*
0	NUMPARSET		Number of PARSET parameters NOT including NUMPARSET and PARVERSION (used internally)
1	PARVERSION		PARSET version number (used internally)
2	BINSCALEXP		Energy scaling factor; internally calculated
3	BINSCALE		Energy scaling factor, proportional to SLOWLEN
4	FIPSCALE		Controls FiPPI energy scaling (depends on SLOWLEN)
5	BLFILTER		Baseline filter average length = 32768/BLFILTER
6	SLOWLEN		Slow Filter Ramp length
7	SLOWGAP		Slow Filter gap
8	PEAKINT		Minimum spacing between pulses
9	FASTLEN		Fast filter ramp length
10	FASTGAP		Fast filter gap
11	THRESHOLD		Fast filter threshold corresponding to the current GENSET.
12	MINWIDTH		Minimum time above threshold for fast filter
13	MAXWIDTH		Maximum time above threshold for fast filter (detects fast filter pileup)
14	SLOWTHRESH		Energy filter threshold corresponding to the current GENSET (only needed for light elements – normally set to 0 to disable)
15	BASETHRESH		Intermediate filter threshold corresponding to the current GENSET. Needs to be set properly to get good baselines.
16	BASTHRADJ		Parameter used for future auto adjustment of BASETHRESH
17	PEAKSAM		Energy Sampling point
18-22	GAINTWEAK0-GAINTWEAK4		If non-zero, this per-GENSET parameter modifies the GAINBASE setting in the GENSET to arrive at

			the GAINDAC setting. E.g. If GENSET 3 is in use, GAINDAC = GAINBASE + GAINTWEAK 3.
23-27	THRESHOLD0-THRESHOLD4		Per-GENSET fast filter threshold values.
23-27	BASETHRESH0-BASETHRESH4		Per-GENSET baseline filter threshold values.
23-27	SLOWTHRESH0-SLOWTHRESH4		Per-GENSET slow filter threshold values.

Appendix D. MicroDXP Hardware Specification

This section describes the first steps that should be taken to design hardware for a system incorporating the microDXP. XIA engineers will provide limited assistance with the actual design, depending on the support agreement.

Board Dimensions and Mounting

The microDXP measures 3.375” x 2.125”, as shown in Figure. C.1, with 0.120” non-plated mounting holes inset by 0.175” symmetrically with respect to each of the four corners. These mounting holes are intended for use with 4-40 or equivalent screws. An alternate board form factor includes 0.1875” blank PCB rails on the two long sides. The rails were included for systems where the microDXP board is to be mounted in a slot. The overall dimensions for the slot-mounted board are thus 3.375” x 2.500”.

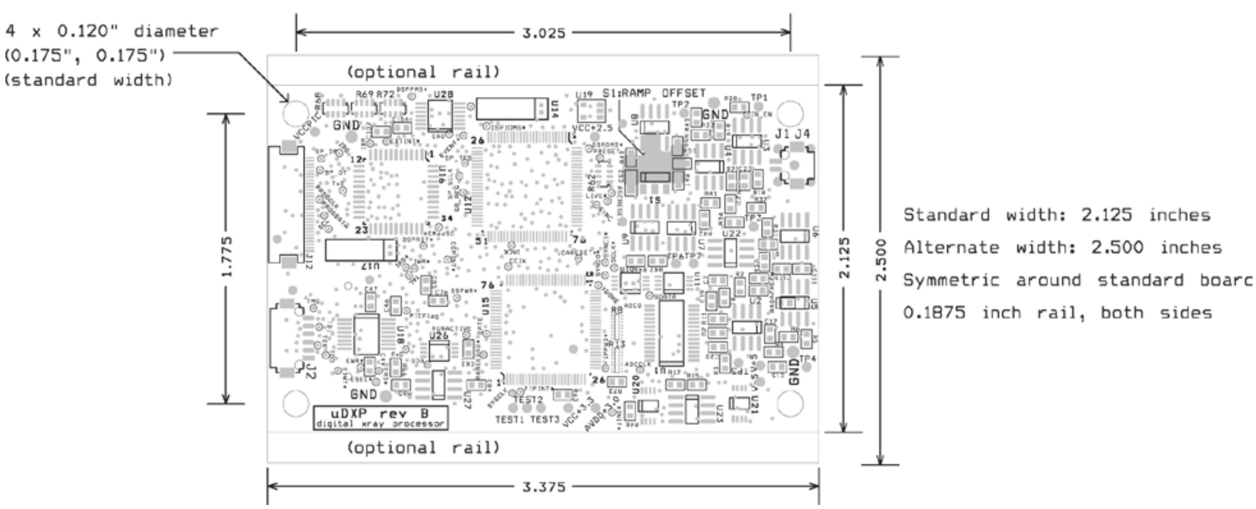

Figure C.1: Dimensions and mounting hole locations.

Preamplifier Type Selector Switch

The only hardware setting on the microDXP board is the preamplifier type selector switch. The location of the miniature two-position slide switch **S1** is displayed in Figure C.2. The two positions are silkscreen-labeled **RAMP** and **OFFSET**, for reset-type and RC-feedback preamplifiers, respectively

Connector Locations and Pinouts

Two connectors carry all electronic signals to and from the microDXP standard assembly, as depicted in Figure C.2. A 2-conductor, 1.25mm pitch connector carries the analog signal from the preamp. The mating connector is a crimp-type socket that accommodates 26-30AWG stranded wire.

A single 30-conductor, 0.5mm pitch flat-flex interconnect carries all communications, power and auxiliary I/O to and from the microDXP. The flex cable provides for two dimensions of freedom, but does require alignment along the axis that bisects all of the contacts. The most likely error would be misalignment of this interconnect, or a reversal of the pinout. Table C.1 details the pin assignments of the flex interconnect.

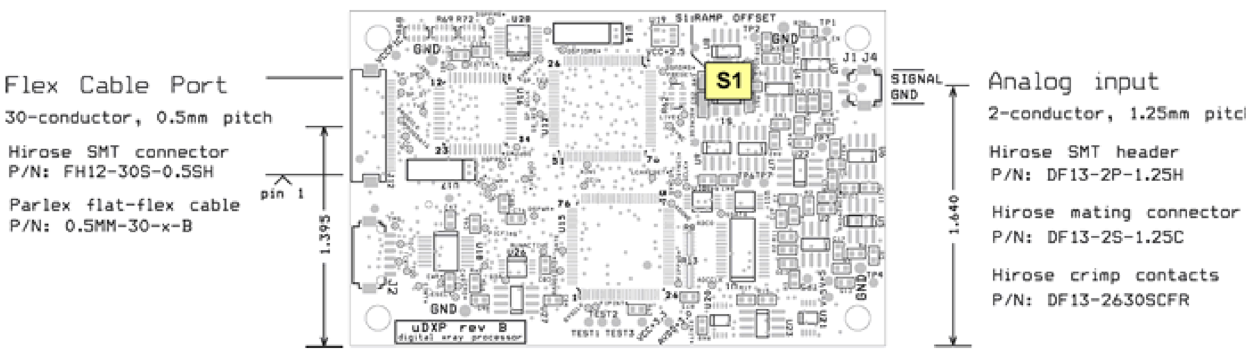

Figure C.2: Connector locations and part numbers (top side).

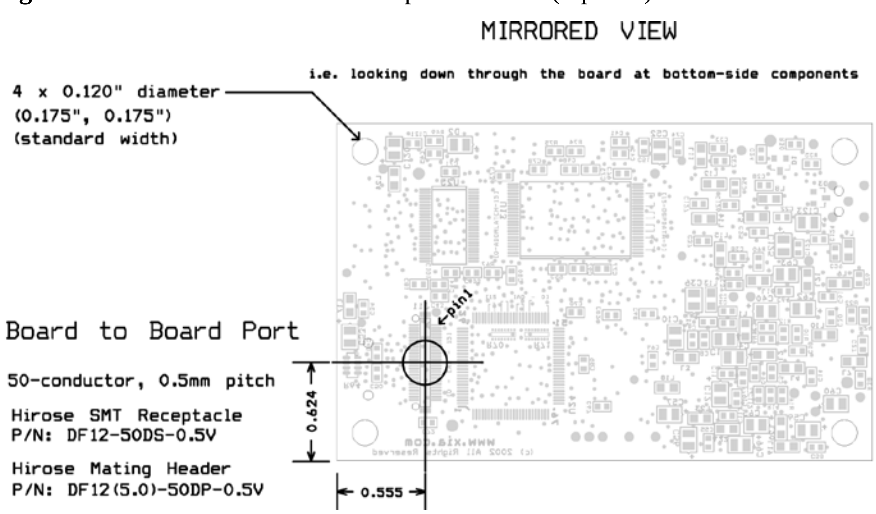

Figure C.3: Connector locations and part numbers (bottom side).

J1 - Analog Input: 2-pin compact right-angle header; or thru-hole LEMO *(Note 1)*. **Hirose P/N: DF13-2P-1.25H (mating P/N: DF13-2S-1.25C; crimp contact P/N: DF13-2630SCFR)**		
Pin #	**Name**	**Description**
1	SIGNAL	Preamplifier output signal
2	GND	Internal ground connection

Note 1: Customers wishing to use an SMA input cable are advised that this capability is provided by means of the MicroComU companion board. See the separate MicroComU Technical Reference Manual. Also contact XIA in order to get quotes for the appropriate variant of the microDXP board.

J12 – Flex Cable Port: 30-conductor, 0.5mm locking flex-cable connector; carries power, communications and auxiliary digital I/O **Hirose P/N: FH12-30S-0.5SH (e.g. flat-flex cable, Parlex P/N: 0.5MM-30-x-B)**		
Pin #	**Name**	**Description**
1	+AVDD	Positive DC supply voltage for analog signal conditioner: Regulated +5.0V; or unregulated +5.5V if on-board regulator present.
2	-AVSS	Negative DC supply voltage for analog signal conditioner: Regulated -5.0V; or unregulated -5.5V if on-board regulator present.
3	GND	Internal ground connection
4	+3.3VCC	+3.3V DC supply for on-board digital components.
5	+3.3VCC	+3.3V DC supply for on-board digital components.
6	GND	Internal ground connection
7	SDA	I^2C data line
8	SCL	I^2C clock
9	ExtInt*	External interrupt line, active low.
10	Gate*	Inhibits data acquisition, active low.
11	GND	Internal ground connection
12	RX	RS-232 microDXP receive (host→microDXP)
13	TX	RS-232 microDXP transmit (microDXP→host)
14	GND	Internal ground connection
15	Vprog	PIC programming voltage
16	ProgData	PIC programming data line
17	ProgClk	PIC programming clock
18	Aux0	Auxiliary configurable digital I/O line: connects to FiPPI
19	Aux1	Auxiliary configurable digital I/O line: connects to FiPPI
20	GND	Internal ground connection
21	Aux2	Auxiliary configurable digital I/O line: connects to FiPPI
22	Aux3	Auxiliary configurable digital I/O line: connects to FiPPI
23	+3.3VCC	+3.3V DC supply for on-board digital components.
24	SPORT_CLK	DSP serial port clock line (ADSP218x SPORT)
25	GND	Internal ground connection
26	SPORT_TDATA	DSP serial port transmit data line (ADSP218x SPORT)
27	SPORT_TFS	DSP serial port transmit frame sync line (ADSP218x SPORT)
28	GND	Internal ground connection
29	SPORT_RDATA	DSP serial port receive data line (ADSP218x SPORT)
30	SPORT_RFS	DSP serial port receive frame sync line (ADSP218x SPORT)

Table D.1: Pin assignments for the 30-conductor flat-flex interconnect.

J11 – Board-to-Board Port: 50-conductor, 0.5mm mezzanine board-to-board receptacle; carries power, communications and auxiliary digital I/O **Hirose P/N: DF12-50DS-0.5V (microCOM / MicroComU mating header P/N: DF12(5.0)-50DP-0.5V)**		
Pin #	**Name**	**Description**
Odd-numbered pins (top to bottom along the right-side of the connector as shown in Figure C.3)		
1	+3.3VCC	+3.3V DC supply for on-board digital components.
3	+3.3VCC	+3.3V DC supply for on-board digital components.
5	+3.3VCC	+3.3V DC supply for on-board digital components.
7	GND	Internal ground connection
9	EAD15	IDMA data/address I/O line (MSB)
11	EAD14	IDMA data/address I/O line
13	EAD13	IDMA data/address I/O line
15	EAD12	IDMA data/address I/O line
17	EAD11	IDMA data/address I/O line
19	EAD10	IDMA data/address I/O line
21	EAD9	IDMA data/address I/O line
23	EAD8	IDMA data/address I/O line
25	EAD7	IDMA data/address I/O line
27	EAD6	IDMA data/address I/O line
29	EAD5	IDMA data/address I/O line
31	EAD4	IDMA data/address I/O line
33	EAD3	IDMA data/address I/O line
35	EAD2	IDMA data/address I/O line
37	EAD1	IDMA data/address I/O line
39	EAD0	IDMA data/address I/O line (LSB)
41	GND	Internal ground connection
43	EWR*	IDMA write strobe (Active LO)
45	ESel*	IDMA device select INPUT (must be asserted LO for IDMA use)
47	ERdy*	IDMA data ready (Active LO) OUTPUT
49	ERD*	IDMA read strobe (Active LO)
Even-numbered pins (top to bottom along the left-side of the connector as shown in Figure C.3)		
2	+AVDD	Positive DC supply voltage for analog signal conditioner: Regulated +5.0V; or unregulated +5.5V if on-board regulator present.
4	-AVSS	Negative DC supply voltage for analog signal conditioner: Regulated -5.0V; or unregulated -5.5V if on-board regulator present.
6	+3.3VCC	+3.3V DC supply for on-board digital components.
8	GND	Internal ground connection
10	SPORT_RFS	DSP serial port receive frame sync line (ADSP218x SPORT)
12	SPORT_RDATA	DSP serial port receive data line (ADSP218x SPORT)
14	GND	Internal ground connection
16	SPORT_TFS	DSP serial port transmit frame sync line (ADSP218x SPORT)
18	SPORT_TDATA	DSP serial port transmit data line (ADSP218x SPORT)
20	GND	Internal ground connection
22	SPORT_CLK	DSP serial port clock line (ADSP218x SPORT)
24	GND	Internal ground connection
26	Aux3	Auxiliary configurable digital I/O line: connects to FiPPI
28	Aux2	Auxiliary configurable digital I/O line: connects to FiPPI
30	Aux1	Auxiliary configurable digital I/O line: connects to FiPPI
32	Aux0	Auxiliary configurable digital I/O line: connects to FiPPI
34	Gate*	Inhibits data acquisition, active low.
36	SCL	I^2C clock
38	SDA	I^2C data line
40	GND	Internal ground connection

42	RX (B)	RS-232 microDXP receive (host→microDXP)
44	TX (B)	RS-232 microDXP transmit (microDXP→host)
46	GND	Internal ground connection
48	EA/D*	IDMA address (HI) / data (LO) selector INPUT
50	ExtInt*	External interrupt line, active low.

Table D.2: Pin assignments for the 50-conductor board-to-board interconnect.

Power Supplies

Note: *Excessive ripple on the analog supplies (>20mVpp) can seriously degrade system performance. If +/-5.0V is supplied directly, either linear regulated or high-quality switching supplies should be used.*

The microDXP requires three supply voltages to operate. A supply voltage of +3.3V is used to directly power most on-board digital circuitry, with minimal LC filtering at the board entry point. On-board voltage regulators also generate from this supply +3.0V for the ADC and +2.5V for the DSP. The total current requirement depends on the selected clock speed, ranging from 80mA to 130mA. The ripple requirements for this supply are not particularly stringent, though excessive radiated noise is to be avoided. If a switching supply is used, it should be well shielded from the microDXP.

Supply voltages of +/-5.5V are regulated on-board by default to generate +/-5.0V to power the analog components. The microDXP is thus intended to tolerate some conducted EMI (<100mV pk-pk) from switching supplies. The +/-5V analog regulators can be bypassed, and thus a slight reduction in power achieved, if low-noise (ie. linear or carefully designed switching) supplies are used. If the regulators are bypassed, only minimal LC filtering will be applied at the board entry point.

Current draw on the analog supplies is dominated by the optional variable-gain amplifier, which alone draws 12.5mA. The remaining analog circuitry draws roughly 10mA. The total required current is, conservatively, 15mA without variable-gain; 30mA with variable gain.

XIA has recently introduced the MicroComU companion board to complement the microDXP. The MicroComU will generate all voltages required by the microDXP, at the required currents and noise performance. The MicroComU/microDXP board set may be powered off the USB bus or off a single external AC power adapter. Customers wishing not to have to worry about power supply design are advised to purchase MicroComU companion boards.

Regulated Supply Option: (<20mV pk-pk noise)			
Voltage Range	*Current (min)*	*Current (max)*	*Description*
+3.3V +/- 150mV	100mA	130mA	Decent switching supply
+5.0V +/- 100mV	25mA	30mA	Linear or high-quality switching
-5.0V +/- 100mV	25mA	30mA	Linear or high-quality switching
Unregulated Supply Option (<100mV pk-pk noise)			
Voltage Range	*Current (min)*	*Current (max)*	*Description*
+3.3V +/- 150mV	100mA	130mA	Decent switching supply
+5.5V to +6.0V	25mA	35mA	Decent switching supply
-5.5V to –6.0V	25mA	35mA	Decent switching supply

Table D.3: Power supply specifications for the microDXP.

Clock Speed [MHz]	*Voltage Supply*	*Current [mA]*	*Power [mW]*	*Comment*
8	VCC	89.7	296.0	+3.3V digital – includes ADC
8	V+	20.3	101.5	+5V analog – includes VGA*
8	V-	23.3	116.5	-5V analog– includes VGA*
			514 mW	**Total power consumption at 8MHz**
16	VCC	103.0	339.9	+3.3V digital – includes ADC
16	V+	20.3	101.5	+5V analog – includes VGA*
16	V-	23.3	116.5	-5V analog– includes VGA*
			557.9 mW	**Total power consumption at 16MHz**

Table D.4: Power consumption by pipeline clock speed.

*****The (optional) variable gain stage draws approximated 12.5mA.

Appendix E. RS-232 Communications

This appendix describes the basic microDXP RS-232 command and response protocol. Please refer to the **RS-232 Command Specification** (a separate document) for a detailed presentation of all RS-232 commands. DXP-related documents are available online at:

`www.xia.com/DXP_Resources.html.`

The general structure for commands and responses is as follows:

[Esc][Command][Ndata (2 bytes)][data1]…[dataN]**[xor CS]**

where:

`[Esc]`	Escape (ASCII 0x1B) as a command start byte
`[Command]`	Single byte for command number, allowing up to 255 commands. See the tables below for command definitions.
`[Ndata]`	Number of data bytes to follow. Two bytes, low byte first.
`[data1]...[dataN]`	The data bytes.
`[xor CS]`	Exclusive-or checksum (bitwise xor of all bytes except for the initial `[Esc]`). If the checksum is not correct an error response is returned.

The format of responses echo the format of commands; that is, they start with the `[Esc]`character and pass back the command # to which it is responding, followed by appropriate data and checksum. The first data byte of all responses is the return status, which is zero for a successful command. In case of an error, only the error byte is returned – no other data bytes are sent.

The command for starting a run is given below as an example:

Command:

`0x1B`	(the escape character)
`0x00`	(the command)
`0x01`	(the low byte that sets the number of data bytes)
`0x00`	(the high byte that sets the number of data bytes)
`0x01`	(the data byte: new run, clear the data)
`0x00`	(bitwise XOR—excludes escape character)

Response (if successful):

`0x1B`	(the escape character)
`0x00`	(the command is always returned)
`0x03`	(the low byte that sets the number of data bytes)
`0x00`	(the high byte that sets the number of data bytes)
`0x00`	(status is ok)
`0x0B`	(the low byte of the new RUNID=0x1B=27)
`0x10`	(the high byte of the new RUNID=0x1B=27)
`0x18`	(bitwise XOR—excludes escape character)

Response (if unsuccessful):

`0x1B`	(the escape character)
`0x00`	(the command is always returned)
`0x01`	(the low byte that sets the number of data bytes)
`0x00`	(the high byte that sets the number of data bytes)
`0x01`	(status indicates an error)
`0x18`	(bitwise XOR—excludes escape character)